Beyond the Spectrum

Prosodic, Noise-Enhanced and Self-Supervised Features for Speech Recognition

Guillermo Cámbara

TESI DOCTORAL UPF / ANY 2023

DIRECTORS DE LA TESI

Dra. Mireia Farrús i Dr. Jordi Luque

DEPARTAMENT DE TECNOLOGIES DE LA INFORMACIÓ I LES COMUNICACIONS

Acknowledgements

Many years ago, in one of these strange turns of life, my grandpa got to meet a former minister. He, who was born in a town in Castilla (not La Mancha though), whose name I do not wish to recall, would often joke saying that he never thought he would shake hands with a minister. Today, he would certainly be surprised to know that he was taking a future doctor to school, hand in hand! *De los buenos manantiales se forman los buenos ríos* – from good springs come good rivers. Because of that, my first thoughts are with my family. It is not for me to say whether I am good or not, but what I can say is that, thanks to them, I have been able to flow through life.

Maria, with you I've learned that the doors to some of life's greatest treasures are found at the end of long and winding roads. We were once sitting in a cafeteria in L'Hospitalet when you gave me the final push to start my master studies. Five years later, I'm becoming a doctor, and also your family! Thank you wholeheartedly, you know I admire you. And thanks to your family (now mine too!), who have been supportive, loving and caring during all these years.

Speaking about expanding families, this one goes to my "doctoral parents", Mireia and Jordi! I have been given so many things from you, through your wisdom, patience and guidance. You nurtured every step of this process, from the first line of PyTorch code I did in the master thesis to the last comma you adviced me to correct in this manuscript. You're great Jedi mentors, and I'm honoured to be your padawan.

Many people describe the PhD as a lonely journey. Indeed, the PhD student is responsible of making that journey happen, but honestly I don't agree that it lonely at all, I actually think it is quite a collective one. I have been lucky enough to learn from many people at different institutions, so let me thank them all.

To all the TALNers, for being this warm, kind and familiar group that I like to call home. Thanks to Leo, Horacio and (formerly) Mireia for fueling TALN with great humans: Ahmed, Jens, Roberto, Giorgia, Euan, Nico, Santi, Alba, Laura(s), Piotr, Alex Shvets, Joan(s), Mónica, Hamza... Special thanks to Alex Peiró, for the crazy research ideas and friendship that the Fornet's cappuccinos fostered, these are the strongest caffeinated drinks in Poblenou for sure! Beyond TALN, thanks to Aurelio, Lydia, and all the fellows that contribute to the good health of the DTIC department. Thanks to Rodolfo and Ariadna, for their commitment to their master theses, I learned a lot from you!

I have almost been an eternal intern, half of my PhD has occurred at internships. Many thanks to Jordi (once again) and Carlos, from Telefónica Research, I keep fond memories of all the knowledge and funny moments we have

shared. Thanks to you I've met amazing talents (and persons) like Benet, David Bonet, Mariona, Fer and Martin Kocour. You're really great energizers of the Deep Learning community in Barcelona, for which I cannot forget mentioning Xavi Giró, Toni Bonafonte and the rest of colleagues that keep the flame alive in this side of the Mediterranean.

Want more internship appreciations? Here is to Joan, Santi and Jordi, from which I feel grateful to have worked with and been the witness of so many high quality deep dives. They are a trio of audio research rockstars, so being there with them felt like attending the rehearsals of bands like Cream or The Jimi Hendrix Experience.

This road has not finished before the next one has started. For so, I want to send a sincere appreciation to all the Amazon folks that I've met during the last year of my thesis. Thanks to Elena and Vincent, who have supported me as an intern and as a full-time employee. Many thanks too to all my team mates for their kindness and brilliance: Alessandro, Arent, Mateusz, Arnaud, Patrick, Ravi, Mikołaj and company.

Finishing the PhD studies means completing the last educational level in our country. I started my official education path when I was a three year old kid, and I wrap it up twenty-seven years later. That is one of many thousand valid pathways one can take to contribute to their society, no better nor worse than others. But this is the way that I felt would suit me, and it is a costly one that takes a lot of slow cooking and human dedication. At the beginning of this lengthy acknowledgements section (indeed, you see this has been quite a collective journey), I was talking about good springs and good rivers. Let me close it by mentioning the last, but not less important, spring. This is the spring that emanates from all the teachers I have had since childhood until now, with a special affection to the ones I had in Bellvitge. Thanks to all of them. And to all the educators in our country, let me not just give them an acknowledgement, but my sincerest wish that they can get the resources they need to fulfil their commitment to the fullest. I hope I can transform this wish into action some day.

Abstract

Speech recognition, the cornerstone of AI voice assistants, has seen significant improvements due to recent deep learning advancements, often being trained with large labeled data or using large encoders pretrained with self-supervised objectives on extensive unlabeled datasets. Despite these advancements, many deployed systems face suboptimal conditions due to factors like shortage of transcribed speech data, limited computational capabilities on smaller devices, or the challenge of recognizing speech in noisy environments. This dissertation delves into three research avenues for enhancing speech recognition under these challenging conditions, all of them focusing on feature enhancement and extending beyond the conventional use of spectral features.

Firstly, we explore the potential of incorporating prosody and voice quality features into spectral feature-based models in data-limited environments, finding that additional pitch and voice quality measurements significantly reduce word error rates. Secondly, we address noisy environments by assessing the impact of speech denoising on wake-up word detection and propose a joint training approach for speech enhancement and detection models, which improves system robustness in both noisy and clean speech conditions. Lastly, focusing on small-footprint devices, we leverage phonetic information from models like wav2vec 2.0 to enhance keyword classifiers without extra computational load, achieving superior performance and efficiency. This approach, further optimized through k-means clustering for weight compression, ensures faster inference with minimal accuracy loss, demonstrating a synergistic integration of advanced techniques to refine speech recognition technology.

In conclusion, this research addresses critical challenges in speech recognition through three different innovative approaches: (1) we enhance spectral feature-based models with prosody and voice quality in data-limited settings, (2) explore joint training methods for robust performance in noisy environments and (3) optimize keyword classifiers on resource-constrained devices. Thus, after exploring these three avenues, we have not only made advances in each area but also provided a suite of techniques for speech feature enhancement that can be used complementarily to offer robust, adaptable solutions for the diverse challenges faced by contemporary AI voice assistants.

Keywords: speech recognition, speech enhancement, self-supervised learning, prosody, voice quality features

Resum

El reconeixement de la parla, que és la peça central dels assistents de veu per IA, ha vist notables millores degut als avenços recents en tècniques d'aprenentatge profund. Aquestes prioritzen habitualment l'entrenament amb grans bases de dades etiquetades o bé fan servir enormes codificadors que es pre-entrenen amb dades extensives sense etiquetar. Tot i aquests avenços, molts sistemes operen en condicions subòptimes a causa de factors com l'escassetat de dades, capacitats de computació restringides en dispositius petits, o l'alta presència de soroll a l'ambient. Aquesta dissertació recorre tres línies d'investigació per millorar el reconeixement de la parla davant d'aquestes condicions dificultoses, totes elles enfocades en la millora de les característiques extretes de la parla, anant més enllà de l'ús convencional de característiques espectrals.

Per començar, explorem el potencial de complementar les característiques espectrals amb característiques prosòdiques i de qualitat de la veu, en sistemes amb limitacions de dades d'entrenament, trobant millores significatives en la reducció d'errors en la transcripció de la parla. A continuació, adreçem el reconeixement en ambients sorollosos, primer mesurant l'impacte de l'ús de models de neteja del soroll sobre models de detecció de paraules d'activació, i després proposant un mètode d'entrenament conjunt pels models de neteja i detecció, incrementant la robustesa del sistema total en condicions silencioses i sorolloses a l'hora. Finalment, centrant-nos en petits dispositius de computació limitada, aprofitem la informació fonètica de models com wav2vec 2.0 per millorar els classificadors de paraules clau sense càrrega computacional addicional, aconseguint un rendiment superior i millor eficiència. Aquest enfocament, optimitzat encara més mitjançant l'agrupament k-means per a la compressió de pesos, assegura una inferència més ràpida amb una mínima pèrdua de precisió.

En conclusió, aquesta recerca aborda reptes en el reconeixement de parla a través de tres enfocaments innovadors diferents: (1) millorem els models basats en característiques espectrals amb característiques de prosòdia i qualitat de veu en entorns amb dades limitades, (2) explorem mètodes d'entrenament conjunt per a un rendiment robust en entorns sorollosos i (3) optimitzem classificadors de paraules clau en dispositius amb recursos limitats. Així, després d'explorar aquestes tres vies, no només hem fet avanços en cada àrea, sinó que també hem proporcionat un conjunt de tècniques per a la millora de característiques de la parla que poden ser utilitzades de manera complementària per oferir solucions robustes i adaptables als diversos reptes que afronten els assistents de veu IA moderns.

Paraules clau: reconeixement de la parla, millora de la parla, aprenentatge auto-supervisat, prosòdia, paràmetres de qualitat de veu

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Speech recognition is a domain of speech technology focused on transcribing spoken words into text. Speech recognition models are paramount for the correct behavior of voice assistants, as the accuracy of their transcriptions is necessary for proper understanding of the intent of a user. Transcribing is challenged by the infinite amount of variability that speech recordings have: different speaker identities, accents or noisy conditions, for instance. Tackling this issue has been done in the latest years with Deep Learning (DL) models. These systems excel at refining input audio into semantically rich latent features, that are easier to classify into text. DL models are able to abstract away all the acoustic information that is independent of the words pronounced (speaker traits, background noises, etc.), all in an automatic manner, with little to none hand engineering. However, their success in doing such task is heavily conditioned by the amount of transcribed speech available for training, as well as how well such data fits the use case of the domain and the number of parameters that the models have. Thus, part of the research in speech recognition during the recent years has addressed the data scarcity issue, finding ways to enrich audio features in order to make the most even with little data. Plus, there is a research focus on small models that deliver fast and robust predictions for small footprint devices, like those that operate offline in electronic devices. The aim of this thesis is to follow up on such research, exploring three complementary techniques of audio feature enhancement for speech recognition in settings with low data resources and small footprint devices. Particularly, we delve into these feature techniques by assessing different speech recognition tasks: (1) large vocabulary speech recognition, for which we will refer simply as automatic speech recognition (ASR), (2) keyword spotting (KWS), which is the task of recognizing a smaller set of words or phrases and (3) wake-up word (WUW) detection, which is a subset of KWS where the single word or phrase that triggers a device has to be detected. Thus, the motivation behind this thesis is to answer the following research questions:

I. **Could the incorporation of prosody and voice quality characteristics contribute to enhancing existing spectral feature-based speech recognition models trained in low resource scenarios, with less than 1000 hours of data?**

II. **How does the application of speech denoising affect the task of wake-up word detection, at different noise levels? Plus, can we improve the way speech enhancement and wake-up word detection models are jointly trained?**

III. **Is there any method for using the phonetic information embedded within an acoustic model based on self-supervised learning such as wav2vec2.0, that enhances speech recognition, all while avoiding extra computational burden and latency?**

1.1. Motivation

Deep learning has become the cornerstone of many AI-related applications in recent years, speech technology being no exception. For instance, fully convolutional and Transformer-based models are capable of providing transcriptions from raw audio containing speech in an end-to-end fashion (Zeghidour et al., 2018c; Graves et al., 2006; Wang et al., 2020c). Furthermore, this task may be carried out under very diverse contexts: different speakers, accents, background noises, etc. Models need large amounts of data and parameters in order to generalize well to such variations, since they are designed mainly to output tokens like graphemes or phonemes given a certain audio, without any explicit knowledge on emotions, acoustic scenes, accents or voice pathologies (Moore, 2003). Abundance of labeled data and computation resources is far from reality for many use cases and languages where an ASR module is needed. Working under low-resource situations is common, whether working for a language with limited public corpora, or for a use case with a specific domain that does not have much labeled data, or simply working on small footprint devices with limited computational capacity that must face acoustically challenging environments. Thus, we aim to conduct research on different venues for these variety of challenges, tackling feature enhancement for scenarios of data scarcity, high environmental noise and computational constraints separately. We explain in detail the motivations behind our three research questions in subsections 1.1.1, 1.1.2 and 1.1.3.

1.1.1. Speech Recognition with Prosody and Voice Quality Features

Prosody is undoubtedly crucial in human communication and provides layers of meaning to spoken language, which might not be immediately evident from the mere textual representation of the words. Prosody encompasses elements like intonation, rhythm and stress. When it comes to transcribing speech audio into text, most ASR systems focus primarily on recognizing phonemes and words rather than the prosodic aspects of the speech. However, this does not mean that prosody does not play a role in speech recognition. To begin with, it is an additional factor of acoustic variance that the model needs to disentangle, as ASRs need to be consistent to different speech rhythms, intonations and accents, for instance. Furthermore, prosody is not only something to be abstracted away at speech recognition, but it also yields clues that help in the transcription process. For instance, proper understanding of stress in syllables helps disambiguating the meaning of homographs (words that are written the same but have different meanings), which is important for knowing the context of a sentence. Plus, prosodic information can aid in finding where to place punctuation marks in transcriptions, especially in determining sentence and word boundaries or question marks (for rising intonation).

In the past, some ASR proposals would use explicit information from pitch features to help disentangling the effects of prosody when transcribing speech (Povey et al., 2011; Guglani and Mishra, 2020; Magimai-Doss et al., 2003). Other works would also use complementary voice quality features to enhance the tasks of speaker recognition (Farrús et al., 2007) and diarization (Zewoudie et al., 2014). In (Campbell and Mokhtari, 2003), the authors demonstrate that voice quality attributes serve as important indicators for conveying paralinguistic information. Furthermore, the authors argued that these attributes should even be regarded as prosodic carriers, like intonation and duration, for instance.

In the current context, there is a trend to make systems as end-to-end as possible, mainly working with spectral features or even from the raw waveform only, without additional speech features like prosody ones. It is common to rely on increasing amounts of data and parameters, to learn to handle the implicit information contained in these raw features, such as prosody or speaker identity. However, it is not always possible to work with huge datasets and enough computation for massive models. Recognizing the profound impact of prosody in capturing emotion, intent, and emphasis in human speech — which is paramount for nuanced speech recognition — prompts the pressing question: **can the inclusion of prosody and voice quality features further refine and enrich the prevalent spectral feature-based systems?**

1.1.2. Speech Enhancement for Wake-up Word Detection

Many AI-powered voice devices today operate in environments with ambient noise, ranging from bustling urban settings and busy households to vehicles and outdoor venues. This ambient noise can pose significant challenges to accurate speech recognition. When voice commands are drowned out by extraneous sounds or when the spoken word becomes muddled by overlapping noise, the ability of the devices to discern user intent diminishes. To address this, speech enhancement techniques are employed as a solution. These techniques are designed to filter out unwanted noise, amplify the desired voice signals, and ensure that voice inputs are as clear as possible before they reach the recognition algorithms. By refining the input signal and reducing noise interference, speech enhancement methods play a pivotal role in enhancing the performance and reliability of voice-activated AI devices in real-world noisy conditions.

A particular case of speech recognition is wake-up word detection. Wake-up words or phrases, like ”Hey Siri” or ”OK Google”, are designed to activate voice assistants, and their accurate detection is crucial for optimal user experience. In environments where the wake-up word might be masked by other sounds or overlapping voices, speech enhancement could be the key to ensuring the device responds promptly and accurately. However, we wonder if excessive enhancement might inadvertently distort the voice cues the detection algorithm relies upon. Even more, enhancement on situations without noise may cause artifacts in the audio that would yield to detection errors. This is altogether not a trivial task, since wake-up word models are always activated, waiting to detect the trigger word, so their compute consumption must be low. Thus, this motivated us to study ways to improve the noise robustness of different wake-up word classifiers, in the scenario of a small footprint device recognizing the trigger word ”OK Aura”, an AI voice assistant created by Telefónica[1]. We considered a few questions about how enhancing the noise in the input spectral features might impact optimal performance. **How does the performance of wake-up word detection vary across different signal-to-noise ratios and what is the impact of it when we apply speech enhancement? What are the most effective methods to jointly train speech enhancement and wake-up word detection models?**

[1] https://aura.telefonica.com/es/

1.1.3. Self-Supervised Learning for Small Footprint Keyword Spotting

One of the research venues that has brought the most significant improvements to speech recognition in the recent years has been self-supervised learning. The main idea is to train powerful encoders that extract rich features by defining proxy tasks, where labels are automatically generated from untranscribed data. These features are then used to fine-tune systems for downstream tasks, such as ASR, which require less labeled data compared to traditional supervised learning schemes. Paradigms like wav2vec2.0 (Baevski et al., 2020) or HuBERT (Hsu et al., 2021) have become very popular, being the starting point for fine-tuning and training many voice systems, as they have been widespread in the open-source domain.

However, while wav2vec2.0 and HuBERT have significantly impacted the field due to their performance gains, there is a trade-off in terms of model size and computational demands. These models, by virtue of their architectural depth and complexity, can be resource-intensive. When considering deployment on small footprint devices, such as wearables, smart home accessories, or IoT devices, this presents a challenge. Specifically, speech recognition tasks with smaller vocabulary like keyword spotting, which ideally require real-time responsiveness and low latency, might be affected by the computational overhead of these large models. Hence, while their efficacy is undeniable for server-based applications or scenarios with ample computational resources, more effort is needed to adapt or distill these models for seamless integration into resource-constrained environments. Research has been done in best approaches for model optimization, quantization, and pruning, but we pose the following question: **is there a novel way for leveraging phonetic information encoded in a model like wav2vec2.0, that improves keyword spotting performance without additional computational and latency costs?**

1.2. Objectives of the Thesis

Neural speech recognition for large-vocabulary or keywords depends on a large amount of data and significant computational resources, in order to perform robustly in real world scenarios. In many setups, data are scarce or the models operate on small footprint devices, reducing the ability of models to transcribe effectively, especially in a modality such as speech audio, where the variance introduced by speaker identity, prosody, accents and environmental noises is very high. This poses the motivation behind the main questions addressed in this thesis

which can be summarized as:

I. Could the incorporation of prosody and voice quality characteristics contribute to enhancing existing spectral feature-based systems?

II. What is the relationship between word detection performance and varying signal-to-noise ratios, and how does the application of speech enhancement affect this relationship? Additionally, what are the most successful approaches for co-training speech enhancement and wake-up word detection models?

III. Is there any innovative method for utilizing the phonetic information embedded within a model such as wav2vec2.0 to enhance keyword spotting performance, all while avoiding extra computational burden and latency?

We hypothesize that prosody and voice quality characteristics might still be useful for large-vocabulary ASR, especially in situations where training data available is up to a maximum of 1000 hours of speech. Regarding speech enhancement on wake-up word detection, we argue that both need to be coupled at training, so speech enhancement specializes on cleaning speech to optimize wake-up word detection, without causing distortions on clean signals or erasing speech from very noisy recordings. Lastly, we want to prove that the phonetic information encoded in wav2vec2.0 quantized codebooks can be transferred as weights to novel architectures in the keyword spotting domain.

1.3. Outline of the Thesis

The rest of this dissertation is structured as follows:

- **Chapter 2** explores the evolution of Automatic Speech Recognition (ASR), starting with classic Hidden Markov Models, transitioning to deep learning-based methods, and introducing self-supervised learning paradigms. We further examine the role of prosody in refining ASR and conclude with the significance of Speech Enhancement in improving recognition accuracy in noisy environments.

- **Chapter 3** delves into research on the impact of prosody and voice quality features, such as jitter and shimmer, on ASR performance. We investigate their influence in a setup constrained to a maximum of 1000 hours of data

from the LibriSpeech dataset, to simulate a low-resource language. We experiment with convolutional and Transformer architectures, finding higher impact from prosody and voice quality features on the latter.

- **Chapter 4** looks at optimizing the recognition of the wake-up word "OK Aura" for Telefónica's voice assistant, Aura. We establish a new dataset centered fro "OK Aura" wake-up word and explore the integration of speech enhancement with various small footprint classifiers. By experimenting with different coupling methods of speech enhancement and wake-up word classifiers during training, we discuss valuable insights into best practices for word recognition in noisy scenarios.

- **Chapter 5** investigates the potential of harnessing the phonetic information present in the wav2vec2.0 codebook, which is often discarded post-training, to enhance a keyword spotting model. Our objective is to identify a computationally efficient method to leverage information in such a self-supervised learning encoder. We delve into the natural synergy between this phonetic codebook and the innovative Perceiver architecture, particularly through weight initialization strategies.

- **Chapter 6** presents the concluding remarks of the thesis based on the achieved goals and also suggests potential paths for upcoming research and implementations.

Chapter 2

STATE OF THE ART

Speech recognition modeling has come a long way in recent years, going from classic machine learning techniques like Hidden Markov Models or Gaussian Mixture Models to huge deep learning models that train with even hundreds of thousands of speech audio. The goal of this chapter is to introduce some of the most important milestones in speech recognition modeling, as we build our research on top of some of these models. Furthermore, as our research is focused towards the improvement of ASR models through feature enhancement, we dedicate specific sections to it. Particularly, we summarize advances in the application of the three techniques that we study to enhance features in ASR: prosody addition, noise cleaning through speech enhancement and the use of self-supervised learning representations.

2.1. Automatic Speech Recognition Modeling

In this section, we introduce the state of the art in speech recognition modeling, delving into the historical progress and current advancements of this field. Speech recognition, in essence, is the ability of a machine or program to identify and transcribe human language. It is the technology that powers virtual assistants, transcription services, hands-free computing, and many more applications that are now a part of our daily lives.

We begin our exploration with a look at classic models that have served as the foundational basis for speech recognition. These include the Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs). From these classic techniques, we transition to discussing more recent approaches that have catapulted speech recognition to new heights. These include methodologies that em-

ploy advanced deep learning architectures like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers. CNNs are particularly effective in processing grid-like topology data, RNNs excel at handling sequential data, and Transformers, a model architecture introduced in the seminal "Attention is All You Need" paper (Vaswani et al., 2017), offer the ability to handle long-range dependencies in sequence data more effectively than RNNs. The aim is to provide a comprehensive understanding of how we have advanced from traditional models to leveraging the power of neural networks, marking a significant evolution in speech recognition technology.

2.1.1. Hidden Markov and Gaussian Mixture Models

Before the advent of deep learning and its application to speech recognition, other classic methods have been typically used, which still provide robust performances nowadays. The signal processing pipeline from such statistical learning methods is still used in many algorithms today, except that some elements are swapped by neural models. Being so, understanding these models is paramount to understand the state of the art in current deep learning models, and where are they heading to.

The classic way of transcribing text from audio is to build a statistical system that outputs the most likely text sequence, W^*, given a set of audio features, X.

$$W^* = \arg\max_x p(X|W)P(W) \tag{2.1}$$

Some of the most commonly used audio features are Mel Frequency Cepstral Coefficients (MFCCs) and Short-Time Fourier Transforms (STFTs), specially the first one, which is the preferred choice due to its closeness to human hearing perception (Davis and Mermelstein, 1980). These are extracted directly from raw audio using rolling windows. Such windows typically have frames sizes around 25 ms with strides of 10 ms. Standard usage of MFCCs typically involves 13 MFCC coefficients, and 20, 40 or 80 filterbanks, for example. This is a starting point for many applications, although exact quantities may vary.

Notice how short are these rolling windows, which are in the time range of the occurrence of a phoneme. The mapping between acoustic features and words is done with the so-called acoustic model, which learns a representation of $p(X|W)$ for generative models. Thus, the acoustic model is intended to classify every small audio feature frame as a target text token. Commonly used targets are phonemes (Bahl et al., 1981), graphemes (Killer et al., 2003), or even word pieces (Schuster and Nakajima, 2012). Classic statistical learning models employ both Hidden

Markov Models (HMM) and Gaussian Mixture Models (GMM) for acoustic modeling. The GMM is used for modeling the distribution of features for a phone, and the transition between phones is modeled by the HMM (Rabiner and Juang, 1986). The transition possibilities are endless, but fortunately it is possible to find an optimal sequence in polynomial time with the Viterbi algorithm (Gales et al., 2008). However, decoding takes into account the effect of two more actors: the lexicon, also known as the pronunciation model, and the language model.

On the one hand, the lexicon contains a direct mapping between the target tokens and known words. For example, the word "data" could be mapped to /ˈdeɪ.tə/ in a General American English lexicon. On the other hand, the language model is trained with text data, and outputs the probability of the occurrence of a whole sentence containing the sequence of words W, which is the $P(W)$ part in equation 2.1. In other words, the lexicon helps bridging the gap between phonemes and words, while the language model aids gathering words into likely sentences.

The final goal during training is to efficiently estimate the parameters in the whole model: the transition probabilities in the HMM, and the mean vector and covariance matrix in the GMM. An adaptation of the Expectation-Maximization (EM) algorithm for HMM models is used, which is called the Baum-Welch algorithm or Forward-Backward algorithm. This algorithm was developed in a series of articles by Baum and Welch during the late 1960s and the early 1970s (Baum and Petrie, 1966; Baum and Eagon, 1967; Baum et al., 1970).

The performance evaluation of ASR models of any type is typically done by measuring the Word Error Rate (WER). WER compares the predicted sequence of words with the true one, taking into account word substitutions S, insertions I and deletions D, along with the number of words N in the sequence.

$$WER = \frac{S + D + I}{N} \tag{2.2}$$

2.1.2. Automatic Speech Recognition using Deep Learning

Research in recent years has shown that practically every element in the ASR pipeline can be substituted by a deep learning model, achieving better performance. This applies for the feature extraction process plus the acoustic, pronunciation and language modeling.

To begin with, an example of the impact of deep learning in audio feature extraction is the work by Jaitly and Hinton, which used Restricted Boltzmann Machines (RBMs) for obtaining higher dimensional encodings from audio (Jaitly and Hinton, 2011). Authors hypothesized that traditional low dimension encodings

like MFCCs may lose information useful for classification, and they trained RBMs to extract audio features that maximized classification performance. More recent approaches have involved automatic training of MFCC-like filterbanks with convolutional models, like in (Zeghidour et al., 2018a) or (Zeghidour et al., 2018b). This opens up for the possibility of fully convolutional ASR pipelines like in (Zeghidour et al., 2018c), discussed later on.

Furthermore, training each module independently with different objective functions may cause bad error propagation between modules, so the efforts have been focused in training these jointly, in an end-to-end (E2E) fashion. Two of the most known paradigms in E2E models are the Connectionist Temporal Classification (CTC) (Graves et al., 2006) and the Sequence to Sequence (Seq2Seq) criteria (Chan et al., 2015).

Frequently, speech data is noisy and unsegmented, which hardens the task of collapsing a sequence of phonemes, or target tokens, into the correct word. CTC is an objective function with an associated neural network that allows to train an acoustic neural model without knowing the alignment between the acoustics and the transcriptions. The function sums out all the possible alignments for a grapheme sequence in order to maximize its probability, as can be seen in equation 2.3.

$$p(Y|X) = \sum_{a_t \in A_{X,Y}} \prod_{t=1}^{T} p_t(a_t|X) \tag{2.3}$$

In other words, the CTC conditional probability of the output sequence Y given the input one X is equal to the marginalization over the set of valid alignments, computing the probability for a single alignment step by step. Besides the full token set, a blank symbol is added. The neural network modeling of such function consists of an encoder, formed by several RNN layers, and a softmax. It is trained with the CTC loss, which is defined as the negative log probability of correctly labelling the sentence, as seen in equation 2.4:

$$CTC(Y, X) = -\ln p(Y|X) \tag{2.4}$$

CTC has been paramount to develop E2E models, being the foundation for many of them. Graves and Jaitly proposed and E2E system, outputting words directly from audio using CTC criterion (Graves and Jaitly, 2014). The CTC function is modified to minimize the expectation of an arbitrary transcription loss function, which yields to a direct optimization of the WER. Direct transcriptions are provided without the need of a lexicon or a language model, even though using

them increases the performance notoriously. This outlines one of the drawbacks of CTC, which is that it does not model the interdependencies between the possible outputs, assuming that the label outputs are independent of each other. Such problem is sorted out by augmenting the encoder in the CTC with a recurrent neural network that models such dependencies between output sequences. This is called the Recurrent Neural Network Transducer, or RNN-T (Graves, 2012). A RNN-T based system suitable for streaming ASR, proposed by Rao et. al. (Rao et al., 2017), allows initialization by a separate CTC-based acoustic model and a RNN language model. Besides, it also shows that using word pieces instead of graphemes can be beneficial for model performance, capturing longer context and reducing substitution errors.

Meanwhile, Seq2Seq criterion usage has emerged as well, motivated by the drawback of CTC of assuming the independence between output labels. The kick-start for Seq2Seq ASR models is done mainly by the Listen, Attend and Spell (LAS) model (Chan et al., 2015), which also produces direct transcripts from the audio signal, without assuming independence in the label sequence. Basically, it consists of an encoder RNN, called listener, and a decoder RNN, called speller. The listener accepts filterbanks, and outputs higher level representations from speech. These are taken by the speller, which takes advantage of the attention mechanism described in (Bahdanau et al., 2014) to output the final text sequence prediction. Nonetheless, LAS model also has its limitations. For example, it is not an online model, since it requires the whole input to be processed before producing the transcripts. Besides, the attention mechanism requires every output token to evaluate every input time step, which is computationally costly. However, RNN-T and LAS models have shown to achieve similar performances, and RNN-T models with attention mechanisms have been implemented in order to solve the shortcomings from both of them (Prabhavalkar et al., 2017).

Moreover, the ability for convolutional neural networks (CNNs) to extract features from 2D images (Krizhevsky et al., 2017) has been leveraged for acoustic modeling tasks, since MFCCs and spectrograms can be considered as 2D images as well. This has led to a series of applications of CNN architectures for speech recognition. An example of this is the usage of convolutional layers in the Deep Speech 2 architecture (Amodei et al., 2016), which enhances the performance regarding its predecessor, Deep Speech (Hannun et al., 2014), mainly based on RNNs. In the newer model, CNNs are applied to find correlations in the spectrogram, right before passing the new features to the recurrent layers. This, along other improvements, helped to enhance the performance of the previous model. However, keep in mind that approximately 12000 hours of speech audio are needed for the model to reach its full potential. This is typical for many E2E models, since they need to generalize lots of information, like speaker variability,

acoustic scene sounds, noises, reverberations, accents, etc.

A clear example of the convolutional paradigm is the fully convolutional model proposed by Facebook (Zeghidour et al., 2018c). Such model proves that it is possible to achieve state-of-the-art results with an E2E convolutional model, not only using a convolutional acoustic model, but also a convolutional language model. For feature processing, it uses the convolutional extractor previously mentioned (Zeghidour et al., 2018b), which yields to substantial improvements regarding classic mel-filterbanks, specially when applied to noisy audio. The acoustic model is a convolutional neural network with gated linear units (Conv-GLUs) (Liptchinsky et al., 2017), which is one of the keys for its high performance. GLUs are originally developed for building convolutional language models that provide competitive results, with a reduced latency regarding sequential RNN-based models (Dauphin et al., 2017). Besides, they also have proved to reduce the vanishing gradient problem. Such convolutional acoustic model is trained with the Auto Segmentation Criterion (ASG) (Collobert et al., 2016), which is based on CTC, even though it removes the blank symbol and contains a learnable weight matrix modeling the transitions between letters. The GCNN-14B language model from (Dauphin et al., 2017) is used during decoding for resolving into sentences. This model has 14 convolutional residual blocks with convolutional layers before and after each block, placed in order to ensure computational efficiency (He et al., 2016). An efficient beam-search decoding from (Collobert et al., 2016) is used, which takes into account the predictions from both the acoustic and the language models. One of the most notable outcomes from this research is the creation of wav2letter++ (Pratap et al., 2018), an open-source speech recognition framework provided by Facebook, that contains recipes for many models based on CNNs. To continue with, let's introduce some of the most studied ASR models based on CNNs.

One of the natural problems in speech recognition using lexicons is how to deal with out-of-vocabulary (oov) words. The lexicon-free model from (Likhomanenko et al., 2019) shows that using a convolutional language model trained with characters, instead of words, is very efficient in detecting such oov words. Even more, overall WER is on pair with other baselines using word-based language models. This model uses the Conv-GLU AM from (Collobert et al., 2016).

Another state-of-the-art architecture is the sequence-to-sequence model with time-depth separable (TDS) convolutions from (Hannun et al., 2019). The encoder from such model is constituted by TDS blocks, which are 2D convolutions over time that allow for larger receptive fields, without highly increasing the number of parameters. This way, it is possible to obtain a very optimized network, capable of achieving high accuracies while retaining low latencies. As a matter of fact, it is proven that such architecture is very convenient for streaming online applications

by limiting the future context, in order to keep a low latency without a big impact in WER performance (Pratap et al., 2020). This is the key for performing ASR task in a device with low computational resources, like a smartphone, without depending on the cloud.

On and on, convolutional models seem to be more efficient than RNN, even though the cost of lower latencies comes in the form of slightly lower performances. A model that addresses this problem is ContextNet (Han et al., 2020), which is a novel architecture combining CNN, RNN and transducer architectures. This model implements a convolutional encoder that uses squeeze-and-excitation layers (Hu et al., 2018). These squeeze local filters into single global context vectors, merging back global information to local vectors and finally multiplying both of them. This way, the convolutional filters obtain the information from larger contexts, which is a typical advange of RNN and transformer models. A RNN-T decoder is used after the encoder, and a downsampling scheme is proposed in order to maintain lower latency rates.

Besides models based on CNNs and RNNs, the speech recognition community has adopted the Transformer architecture as well for building acoustic models, achieving state-of-the-art performances (Wang et al., 2020c). The Transformer model (Vaswani et al., 2017) is based on multiple heads of attention mechanisms, allowing to compute representations from both input and output in a parallel fashion. Such model does not need any sequence-aligned RNN nor any sort of convolution in its implementation. However, since its creation, it has been combined with these in order to boost performances. For example, in the work of (Synnaeve et al., 2019), a convolutional front-end is appended to a Transformer-based AM. One of the most recent examples of top performing ASR models is Whisper (Radford et al., 2023), which shows the ability of Transformers for scaling up with data, as it is trained with 680k hours in a variety of speech tasks.

In any case, as the advent of Transformers came close in time to the rise of self-supervised learning models, many of the more popular Transformer-based ASRs also use self-supervised schemes to obtain high-performant feature representations. We look at this in the following section.

2.2. Self-Supervised Learning for Speech Recognition

Deep supervised learning models are capable of achieving astonishing performances, but there is an inherent cost for it, which is labeled data. Robust speech recognition modules require amounts of labeled audio in the order of thousands of hours (Moore, 2003). For many real world use cases, it is not feasible to obtain such quantities of speech audio. Working under low resource conditions is very common, be it because of a language with scarce resources, or because of working with a very specific domain that does not have much available data. This is the main motivation behind research in self-supervised learning models, which are able to leverage information from unlabeled or partially labeled data, or even data from other modalities like text or video. This way, the lack of labeled speech data is compensated with other forms of information.

Proxy predictive tasks are used in order to train self-supervised models able to extract high-level representations from data. For example, ELMo (Embeddings from Language Models) (Peters et al., 2018) representations are commonly used in Natural Language Processing (NLP) applications. These are obtained during training of a language model, and then applied to other challenging linguistic tasks, like sentiment analysis or question answering. Besides NLP, self-supervised learning has also found success in the Computer Vision (CV) field, being VideoBERT and example of it (Sun et al., 2019). There is an interest on applying such self-supervised learning methods to ASR as well, up to the point that there has been a special session for it in InterSpeech 2020 conference. Current approaches include future and mask predictions, generation of contextual data and chaining ASR and TTS.

The future prediction approach consists of training powerful autoregressive models, capable of predicting future samples given input data. For example, the Contrastive Predictive Coding model (CPC) from (Oord et al., 2018) uses a probabilistic contrastive loss, learning the underlying shared information between different parts of the signal. This way, local information like noise is discarded, and further global correlations are extracted for such high-level representation. Other models like Autoregressive Predictive Coding (APC) (Chung and Glass, 2020) show that such predictive representations outperform traditional features like MFCCs, and reduce the needed size of downstream labeled data and model parameters. In the line of these models, the architecture proposed by Facebook AI Research, wav2vec (Schneider et al., 2019), uses a multi-layer CNN for the predictive task. Optimization is done with a noise binary classification task. This means that the system has to predict if whether a future sample is original or not,

given that a true future sample can be substituted by a different distractor sample. Facebook provides the code for training the pre-trained models in the fairseq toolkit (Ott et al., 2019).

Mask prediction is another predictive task, which aims to obtain speech representations by predicting masked parts from an input signal. The Mockingjay model (Liu et al., 2020), for example, is trained to predict the current input signal frame, given the previous and the future contexts. It consists of bidirectional Transformer encoders, and outperforms the use of MFCCs with 100% of data, using only the 0.1% of it. Speech-XLNet (Song et al., 2020) is another mask prediction approach, which is based on XLNet (Yang et al., 2019), used in NLP. By shuffling speech frames from an input signal, the model is forced to learn global structures in data, given the local permutations.

One of the most popular mask prediction models for speech is wav2vec2.0 (Baevski et al., 2020). It consists of a convolutional encoder that compresses the waveform to a latent space, which is then passed through a vector-quantized codebook to learn discrete representations of speech. The authors found that such representations were very similar to phonetic units. Being so, the task of wav2vec2.0 model consists of masking some latent vectors after the convolutional encoder, then passing unmasked and masked latents through a Transformer encoder, and predicting to which vector-quantized codes the masked parts correspond to. In this sense, the task resembles what NLP models like BERT (Kenton and Toutanova, 2019) do, which is prediction of masked words, whereas wav2vec2.0, predicts masked pseudo-phonemes. Soon after its release, it was quickly adopted by the community, with recipes and models available in popular frameworks like HuggingFace (Wolf et al., 2019), SpeechBrain (Ravanelli et al., 2021) or fairseq (Ott et al., 2019). Wav2vec2.0 was also followed by HuBERT (Hsu et al., 2021), a model with a very similar architecture based on Transformers, however it uses k-means clustering to quantize the speech space into discrete units, showing even better scores in different downstream tasks than wav2vec2.0 (Yang et al., 2021).

One final observation regarding masked prediction models is that a potential major limitation is their tendency to be less effective in generative tasks due to their design. This is because these encoders learn to get rid of low-level details in the waveform to focus on the semantics. Thus, models like WavLM (Chen et al., 2022) combine masked prediction with denoising tasks to learn representations that are suitable for handling low and high level details of speech. The idea of multi-task representation learning had been previously explored in PASE (Pascual et al., 2019), which stands for problem-agnostic speech encoder and its most recent version, PASE+ (Ravanelli et al., 2020). PASE consists of a single encoder followed by multiple workers, which specialize in jointly solving different self-supervised tasks. The learned features contain information about the speaker

identity, phonemes and prosodic cues. The mentioned workers are regressors and discriminators. The first ones work with input features from raw audio and the latter ones learn to discriminate between positive and negative samples with binary cross-entropy loss. In other words, the neural encoder, based on the SincNet model (Ravanelli and Bengio, 2018), firstly extracts a higher-level representation from audio, which is fed into each one of the seven proposed workers, being these small feed-forward networks. The four regression workers try to learn the modeling of four different input features, by minimizing the mean squared error (MSE) between the original features and the modeled ones. These features are: the input waveform, the log power spectrum (LPS), the MFCCs and a mix of prosody features (the logarithm of the fundamental frequency, the energy, the probability-of-voicing and the zero-crossing rate). The other three workers are used for binary discrimination, doing different tasks like distinguishing local and global features from distractor samples or predicting which samples belong to the future or the past, given an anchor sample.

Going back to wav2vec2.0 and HuBERT, these are quite large models, easily getting to hundreds of millions of parameters. That is because their big Transformer encoders are used as feature extractors. Still, we think of novel ways to leverage their information for models that require faster inference. In this thesis, we will dedicate Chapter 5 to explore how wav2vec2.0 codebook, which captures latent pseudo-phonetic information, can be used for fast speech recognition. For now, let's have a look at other ways of enriching speech features beyond the spectrum.

2.3. Prosody in Speech Recognition

Prosody concerns linguistic properties beyond phonetic segments, like syllables and larger speech units (suprasegmental), analyzing features like intonation, rhythm or stress. Prosody features like intonation, stress or duration have been used as complements for features like MFCCs or STFTs. The use of such features helps the recognizer to distinguish between questions and answers, or to group phrases properly, for example. Furthermore, some languages like Mandarin Chinese are tonal, which means that the word meaning is affected by tone.

These features have been frequently used for speech recognition with classic systems. For example, Kaldi, a commonly used open-source speech recognition framework (Povey et al., 2011), uses a pitch and probability-of-voicing (POV) extractor algorithm specially suited for ASR (Ghahremani et al., 2014). In such framework, these features can be used to train GMM, HMM and DNN based models. However, since the advent of E2E models, it seems that prosody features

have been pulled off the road, in favor of only spectral features or neural feature extractors. The wav2letter++ framework, for example, does not provide with pitch extraction functionalities up to date.

The current trend lends towards minimal feature extraction, expecting that E2E models automatically learn the effects of prosody by exploring huge amounts of data. Nevertheless, for low-resource situations, it may still be needed to evaluate explicit prosody features, in order to aid models that are not fed with lots of hours. Furthermore, prosody features are used as additional information in self-supervised models like PASE (Pascual et al., 2019), proving that they are beneficial for training. Being so, it is plausible to keep using prosody features as an enhancement for modern models, whether with supervised or self-supervised learning methods. Besides, the speech chain proposal where synthetic samples from TTS are fed to ASR requires of speech variations in the TTS model, to improve the ASR one. As producing variations in synthesized speaker traits has brought improvements, applying different prosodic contours to such synthesized voices could bring performances even further, which is suggested in (Wang et al., 2020b).

Furthermore, prosody features have been used as well for accent detection (Rouas, 2007; Ananthakrishnan and Narayanan, 2007; Bougrine et al., 2018). Since they carry suprasegmental information, they inherently denote information for dialects, which can be leveraged by ASR systems in order to boost performances (Zheng et al., 2005). Specially, since it is acknowledged that accent diversity is a common source of error for ASR modules, some works suggest jointly modeling accent detection and acoustic models (Yang et al., 2018). Public databases like Common Voice (Ardila et al., 2020), contain detailed labels on the accent of every utterance, so further research could leverage such knowledge to properly model accented speech. The intersection between prosody features and accent labels seems to be an appealing way to move forward.

Other features closely related to prosody are voice quality parameters like jitter and shimmer. These are measures of cycle-to-cycle variations of fundamental frequency and amplitude in the speech waveform. They carry information about the speaker's voice quality, which have been proven to assist speaker recognition and verification tasks in (Farrús et al., 2007; Farrús and Hernando, 2009), as well as speaker diarization in (Zewoudie et al., 2014). Such voice quality features could also be used to enhance supervised and self-supervised models, given that they have information about speaker variability, an important factor during speech recognition training.

Thus, the addition of prosody and voice quality features has improved many speech tasks, but these features have been seldomly used in neural-based ASR.

Although one of the main advantages of deep learning is to reduce the efforts in feature extraction, we argue that additional prosody features might still be useful for convolutional and Transformer-based approaches, specially for training setups without huge quantities of data. We will explore these ideas in Chapter 3.

2.4. Speech Enhancement

Speech recognition systems aim to convert spoken language into text, and their performance can be significantly affected by noises in the background. Noises introduce extraneous acoustic variations, which can mask or distort the critical phonetic elements of speech, making it challenging for recognition systems to accurately transcribe spoken words. Furthermore, noises are commonly found in the places where voice assistants are used: homes, restaurants, traffic and so on. How could we improve speech features to make speech recognition systems more robust to noise? One way we explore in this thesis is the use of speech enhancement techniques, which attenuate or even eliminate these undesired acoustic interferences.

Before the advent of deep learning, speech enhancement relied on classic methods like Wiener filtering (Meyer and Simmer, 1997) and spectral subtraction (Yang and Fu, 2005). On the one hand, Wiener filtering estimates the clean speech signal by applying a frequency-dependent gain to the noisy speech, based on the signal-to-noise ratio. On the other hand, spectral subtraction estimates the noise spectrum and subtracts it from the noisy speech spectrum. Despite their success in various setups, these methods introduce artifacts and distortions, especially when noises are non-stationary. Deep learning models, however, are more capable of handling such non-stationary noises, as their design is suitable for modeling complex non-linear relationships and learning from vast amounts of data. Consequently, neural-based methods have replaced the classic ones in the recent years, yielding more natural and clearer speech, even in challenging scenarios.

Speech enhancement models may rely on different types of architectures and features. For instance, there are models based purely on CNNs (Park and Lee, 2016), or others using LSTMs (Weninger et al., 2015) for capturing longer contexts, operating on spectral features. As operating on the spectrogram means discarding some parts of the signal, other proposals have performed speech enhancement directly on the waveform, like speech-enhancing WaveNets (Rethage et al., 2018) or autoencoder models like Denoiser (Défossez et al., 2020). Denoiser consists of a convolutional autoencoder with an LSTM between the encoder and the decoder, having an architecture that is very similar to U-Net (Ronneberger et al., 2015). Overall, many current approaches minimize a regression loss in time or

frequency domains (Park and Lee, 2016; Défossez et al., 2020). However, some models have used generative approaches instead, like SEGAN (Speech Enhancement Generative Adversarial Network) (Pascual et al., 2017), which is similar to Denoiser in its U-Net-like architecture, but uses a GAN approach to discriminate signals cleaned by the generative network from the ground truth signals. Other works have iterated on this paradigm, like a proposal that adds more generators for doing multi-stage speech enhancement (Phan et al., 2020). As new generative designs have appeared, they have been applied to speech enhacement as well, like Flow-based models (Strauss and Edler, 2021) and diffusion ones (Lu et al., 2022; Hu et al., 2023).

In this work, we address the usage of neural speech enhancement for a particular case of speech recognition: wake-up word detection. Previous works have explored improving speech recognition with speech enhancement, like jointly training a mask-based enhancement model with an ASR (Liu et al., 2019), or using a fully batch-normalized architecture to regularize the output distribution changes in the front-end at joint training (Ravanelli et al., 2016). Recent literature shows the incorporation of attention to such pipelines, proposing models where both the enhancement and the ASR models are based on self-attention, being optimized in an adversarial joint training (Li et al., 2021). On and on, joint training of speech enhancement and speech recognition models address the issue that the former may introduce out-of-distribution artifacts that are hard to capture by the latter. In the Chapter 4 of this thesis, we perform an extensive study on how different recognition models are affected by the way we couple speech enhancement models to them. As mentioned before, we do this in the case of recognizing wake-up words in a real-world home voice assistant.

Chapter 3

ENRICHING SPEECH RECOGNITION FEATURES WITH PROSODY

Prosody, concerning linguistic properties beyond phonetic segments such as syllables and larger speech units (suprasegmental), focuses on features like intonation, rhythm, or stress. These prosodic features, including F0 contour or speech and articulation rates, complement spectral features like MFCCs or STFTs in speech recognition. They aid in distinguishing between different speech elements, such as questions and answers, and are crucial in tonal languages like Mandarin where tone affects word meaning. Prosodic elements also convey meaning in non-tonal languages by indicating speaker intent. Historically, prosodic features have enhanced ASR system accuracy in both tonal and non-tonal languages, as seen in frameworks like Kaldi (Povey et al., 2011; Ghahremani et al., 2014). However, end-to-end models have shifted focus towards minimal feature extraction, typically relying on spectral features and not prosodic ones. Despite this, in low-resource scenarios or with self-supervised models like PASE (Pascual et al., 2019), prosodic features still prove beneficial. Additionally, prosody aids in accent detection, improving ASR performance in diverse dialects.

Voice quality parameters, such as jitter and shimmer, while distinct, are intrinsically connected to prosody, providing paralinguistic information alongside pitch and duration (Campbell and Mokhtari, 2003). These features have been valuable in various applications, including style, age, and gender classification (Slyh et al., 1999; Wittig and Müller, 2003), emotion detection (Li et al., 2007), speaker verification, recognition and diarization (Farrús and Hernando, 2009; Farrús et al., 2007; Zewoudie et al., 2014), and health diagnostics (Mirzaei et al., 2018; Benba et al., 2014).

Considering the positive use of prosody and voice quality features in all these tasks, we hypothesized that such features would be beneficial for neural ASR as well. We started by incorporating prosody and voice quality features to a convolutional-based architecture, which yielded slight improvements to the recognition task. As Transformer architectures were taking over many speech tasks at that time, we decided to follow up on the work done, applying voice quality and pitch features to an ASR Transformer architecture. We considered that Transformers would better leverage prosodic information, as these features are suprasegmental in nature, and Transformers' attention mechanism is specially useful for enriching features at larger contexts. This time, not only the overall performance of the model was better than the convolutional one, but also prosody and voice quality features had a major impact on performance. We discuss these two works in detail in the following sections.

3.1. Pitch and Voice Quality Features for Convolutional Speech Recognition

Firstly, we studied the effects of adding pitch and voice quality features such as jitter and shimmer to a state-of-the-art CNN model for ASR. Recent CNN-based ASR approaches (Wang et al., 2017; Synnaeve et al., 2019) have the advantage of having large context windows, without the risk of vanishing gradients like in pure LSTM approaches, and being suitable for online streaming applications, while attaining low word error rate (WER) scores. Furthermore, following the trend of making systems as end-to-end as possible, even fully convolutional neural approaches have been proposed, and shown state-of-the-art performances (Zeghidour et al., 2018c). This fully convolutional architecture takes profit of stacking convolutional layers for efficient parallelization with gated linear units (GLU) that prevent the gradients from vanishing as architectures go deeper (Dauphin et al., 2017). We refer to this model as the Conv GLU model.

Pitch features have been previously used for improving classic HMM and DNN baselines, while jitter and shimmer parameters have proven to be useful for tasks like speaker or emotion recognition. Thus, our first intent was to assess the value of adding pitch and voice quality features, like jitter and shimmer, to the spectral coefficients for a convolutional-based model. To this end, the dimension of the mel-frequency spectral coefficients (MFSC) vector, at the input layer, was augmented by the prosodic and voice quality features, and error rates were reported for both Spanish and English speech recognition tasks. Experiments were carried out by using the Conv GLU model. Such was proposed by (Collobert et al., 2016) within the wav2letter's WSJ recipe (Pratap et al., 2018) and reported

state-of-the-art performances for both LibriSpeech (Panayotov et al., 2015) and WSJ datasets. To the best of our knowledge, this was the first attempt to use jitter and shimmer features within a modern deep neural-based speech recognition.

3.1.1. Methodology

Let's outline how we assessed pitch and voice quality features, including the data used, the feature extraction process, the system architecture employed, and the experiments conducted.

Data

The effect of adding pitch and voice quality features was evaluated by means of the Common Voice dataset in Spanish (Ardila et al., 2020) and the LibriSpeech 100h dataset in English (Panayotov et al., 2015). Common Voice corpus is an open-source dataset that consists of recordings from volunteer contributors pronouncing scripted sentences, recorded at 48 kHz rate and using own devices. The sentences come from original contributor donations and public domain movie scripts and it is continuously growing. Although there are already more than 100 hours of validated audio, we kept a reduced partition of approximately 19.0 h for training, 2.7 h for development and 2.2 h for testing sets. The main criterion for the stratification of such partitions was to ensure that each one had exclusive speakers, while trying to keep a 80-10-10% proportion. Every sample can be down voted by the contributors if it is not clear enough, so we discarded all samples containing at least one down vote, to keep the cherry picked recordings as clean as possible. Afterwards, we tried to keep as balanced as possible the distributions by age, gender and accent. Besides, in order to provide with results for a popular benchmark, the proposal was also assessed with the aforementioned LibriSpeech 100h partition in English, consisting of audio book recordings sampled at 16 kHz.

Feature Extraction

As recommended by wav2letter's Conv GLU recipes[1], raw audio was processed to extract MFSCs, applying 40 filterbanks. This served as our baseline, so on top of it we appended pitch and voice quality related features. From now on, when we talk about pitch features we refer to the following three features: the extracted pitch itself, plus the POV for each frame and the variation of pitch across two frames (delta-pitch). Being so, 40 MFSCs were always computed for each

[1] https://github.com/flashlight/wav2letter/tree/main/recipes/conv_glu

time frame, and if specified by the user in the configuration, the three pitch features (pitch, POV and delta-pitch) would be appended to them, plus jitter relative and/or shimmer relative.

There are various pitch extractor algorithms such as Yin (de Cheveignè and Kawahara, 2002) or getF0 (Talkin and Kleijn, 1995). However, we decided to refactor the Kaldi's one from (Ghahremani et al., 2014) within the feature extractor C++ class from wav2letter. The latter algorithm has been frequently tested along the recent years within a wide variety of ASR tasks. It is inspired by getF0 and finds the sequence of lags that maximizes the Normalized Cross Correlation Function (NCCF). It makes use of the Viterbi algorithm for obtaining the optimal lags and, in our implementation, it applies the logarithm to the pitch values as the only post-processing step. The logarithm compresses pitch values to the same order as the MFSCs, which are compressed by the logarithm as well, thus improving numerical stability later on during the training phase, being more robust to outliars. Subtracting the weighted average pitch during post-processing was discarded, since the reported gains in WER by Kaldi are only of a 0.1%. Shimmer is computed measuring the peak-to-peak waveform amplitude at each period where the pitch is extracted, and then performing the corresponding operations, depending on whether we deal with shimmer dB or shimmer relative (see (Farrús et al., 2007)). With the pitch extracted at each period, the same can be done for jitter absolute and relative, by calculating the fundamental frequency differences between such cycles.

System Architecture

Since our purpose is to study how pitch and voice quality features contribute to a convolutional acoustic model (AM), we used the Conv GLU AM from the wav2letter's Wall Street Journal (WSJ) recipe (Collobert et al., 2016). This model has approximately 17M parameters with dropout applied after each of its 17 layers. The WSJ dataset contains around 80 hours of audio recordings, which is closer to the magnitude of our data than the full LibriSpeech recipe (about 1000 hours). We did not do an extensive exploration of architecture parameters, since it yielded good out of the box results with Common Voice and LibriSpeech 100h data.

Regarding Common Voice's lexicon, we used a grapheme-based one extracted from the approximately 9000 words from both the training and development partitions. We used the standard Spanish alphabet as tokens, plus the *ç* letter from Catalan and the vowels with diacritical marks, making a total of 37 tokens. The *ç* character was included because of the presence of some Catalan words in the dataset, like "Barça". The language model (LM) is a 4-gram model extracted with

KenLM (Heafield, 2011) from the training set. Since most of the sentences were shared across partitions, due to the scripted nature of the dataset, we expected an optimistic behavior after applying such LM. Therefore, we are also reporting results given by another 4-gram LM extracted from the Spanish Fisher + Callhome. The Fisher corpus split is taken from the Kaldi's recipe (Weiss et al., 2017). Decoding across AM, lexicon and LM was done with the beam-search decoder provided by wav2letter (Liptchinsky et al., 2017). Furthermore, in order to assess the capacity of the AM by itself, we also evaluated without LM, choosing the final characters with the greedy best path from the predictions of the AM. For the LibriSpeech evaluation, the lexicon and the language model were the same as provided by wav2letter's Conv GLU LibriSpeech recipe. The lexicon was obtained from the train corpus and the language model is a 4-gram model also trained with KenLM.

Experiments

To perform our assessment on the usage of pitch and voice quality features, we tried 5 different feature configurations:

1. 40 MFSCs only
2. 40 MFSCs + 3 pitch features
3. 40 MFSCs + 3 pitch + 1 relative jitter
4. 40 MFSCs + 3 pitch + 1 relative shimmer
5. 40 MFSCs + 3 pitch + 1 relative jitter + 1 relative shimmer

For each one, we computed WERs on both the dev and test sets of Common Voice. Decodings were performed without LM (NoLM), with both in-domain and out-domain LMs, from Common Voice's LM (CVLM) and Fisher + Callhome's LM (FCLM) databases, respectively. Therefore, we obtained 6 WERs for each one of the 5 feature configurations.

Besides the features, the training configurations for each experiment were the same, all based on wav2letter's WSJ recipe. The inferred segmentation was taken out from wav2letter's Auto Segmentation Criterion (ASG) (Collobert et al., 2016), inspired by CTC loss (Graves et al., 2006). The learning rate was tweaked to 7.3, and was decayed in a 20% every 10 epochs, a tuning done with the dev set. A 25 ms rolling window with a 10 ms stride was used for extracting all the features, jitter and shimmer were averaged across 500 ms windows.

For beam-search decoding, the following settings were tuned with the dev set: LM weight set to 2.5, word score set to 1, beam size set to 2500, beam threshold set to 25 and silence weight set to -0.4. In order to tune them, we did not run an extensive exploration of hyperparameters, but after a shallow search we found these to provide good results for both LMs.

Furthermore, LibriSpeech WER was evaluated with dev-clean/other and test-clean/other partitions, using the same AM training recipe as in Common Voice. As we were scaling with a bigger dataset demanding a higher computational cost, the top three parameter configurations found with Common Voice experiments were selected in order to perform such evaluations. Decoding parameters were taken from wav2letter's LibriSpeech recipe.

3.1.2. Results

Table 3.1 reports the word error rates (WER, %) for each one of the 5 feature configurations, for the proposed decodings of Common Voice's test set, without LM (NoLM), with its own LM (CVLM) and the Fisher + Callhome LM (FCLM). For every evaluated case, the best WER score is always provided by one of the models using pitch features, or pitch with voice quality (jitter + shimmer) features, with gains between 1.38% and 7.36% relative WER points.

For the cases without LM, the model with MFSC and pitch features is the one with the best performance, with a relative gain of 1.83% for the test set, respectively. Additional features on the other models also improve the WER score, except for the case with pitch and shimmer only, which yields worse results across all experiments. On the other hand, decoding with CVLM achieves the best WER scores, when training with all the proposed features together: MFSCs, the 3 pitch features, jitter relative and shimmer relative. A 22.90% WER is obtained for the test set in such scenario. As it was expected, the CVLM improves drastically the predictions, because even though it is obtained from the train partition solely, many sentences are shared with the dev and test sets, due to the reduced vocabulary in this dataset.

A more realistic approach is to decode by using an external LM. The FCLM language model is built from the training partition of the LDC Spanish Fisher + Callhome corpus. Although the LM enrollment is performed with less than 20 hours of audio (approximately 16k sentences), it still yields to a reasonable performance compared to the CVLMs decodings. With respect to the prosodic features, the FCLM beam decoding reaches the lower WER in development by using MFSCs only augmented with pitch features; that is, 37.57% WER. The lowest 42.95% WER score in the test set is given by the combination of all pitch

Table 3.1: WER percentages by augmenting spectral features with prosody and voice quality ones. The results are reported on the Common Voice's test sets, comprising 2.7 hours and 2.2 hours, respectively. Error rates are obtained by using a greedy decoding without language model (NoLM) and by a beam search decoding using a 4-gram LM trained both on the Common Voice's training subset (CVLM) and on the training partition of the Spanish corpus Fisher-Callhome (FCLM).

AM	WER (%)		
Features	**NoLM-Test**	**CVLM-Test**	**FCLM-Test**
MFSC	70.07	24.72	44.20
+ Pitch	**68.79**	24.89	43.18
+ Pitch + Jitter	69.56	23.97	43.26
+ Pitch + Shimmer	77.04	25.10	50.60
+ Pitch + Jitter + Shimmer	69.51	**22.90**	**42.95**

and voice quality characteristics. Once again, the best results in terms of WER are provided by models with pitch features, or pitch features with the combination of jitter and shimmer, showing the potential of pitch and voice quality features to improve the performance of an ASR based on convolutional neural networks.

Nonetheless, it is worth noticing how the use of only pitch and shimmer features yields to worse performance for both AM and AM/LM decoding models. Previous behaviour is depicted in the Figure 3.1, where using only shimmer dramatically affects the training stage of the model, making it worse and slower. However, training with pitch features or with pitch and jitter features seems to help at reaching better WER plateaus and at a faster pace. While jitter is a measure of frequency instability in the wave, shimmer is a measure of amplitude instability. Being so, pitch and jitter characteristics might contribute to MFSCs spectral features with independent information, just by synchronising them in a simple concatenation like the proposed one. However, the inclusion of shimmer, which is related to amplitude –as opposed to the pitch and jitter, related to frequency–, is more likely to be understood as a perturbation throughout the convolutional layers that might difficult the acoustic model training.

Even though, it is interesting to see how if shimmer is coupled with jitter and pitch characteristics altogether, the performance obtained yields to more robust results compared to the baseline and independently of decoding with CVLM and

FCLM language models. Other studies already suggest the correlation between jitter and shimmer by the same index, that is, the Voice Handicap Index (VHI) (Schindler et al., 2009), so the convolutional filters may be finding similar correlations, thus improving mutual information when coupled together with spectral features and promoting such as voice measurements as good feature candidates for enhancing the speech recognition of pathological voices. The latter being an interesting hypothesis to look for further evidence.

The impact of pitch and voice quality features is also reported by LibriSpeech experiments, see Table 3.2 and Figure 3.1, with relative improvements of 2.94% and 2.06% for dev-clean and dev-other, respectively, and about 0.96% and 2.87% for test-clean and test-other. Gains seem to be more consistent for the "other" partitions, where there is more accent and prosody diversity than in "clean" ones.

Table 3.2: LibriSpeech WER values for the three best performing combinations of features proposed in the Acoustic Model (AM) in the Common Voice experiments: MFSC, MFSC + Pitch and MFSC + Pitch + Shimmer + Jitter (shortened as "All"). Decoding is done with a 4-gram LM trained with LibriSpeech train set transcripts.

AM	WER (%)			
Features	**dev-clean**	**dev-other**	**test-clean**	**test-other**
MFSC	10.22	31.59	10.38	34.46
+ Pitch	9.94	30.95	**10.28**	**33.47**
+ All	**9.92**	**30.94**	10.37	33.57

Appending pitch characteristics to MFSCs seems to slightly improve the ASR performance. Among them, MFSC + pitch and MFSC + pitch + jitter + shimmer combinations are the ones that provide the most robust behavior across all the experiment. All assessed features carry prosodic information and might aid the network on complementing the information conveyed by solely the magnitude spectrum. For instance, by helping on reducing the MFSC distortion which appears at the lower frequency region of the spectrum (Yadav et al., 2019). Overall, they help boosting the performance of the convolutional acoustic model for both the different databases and the languages studied in this work.

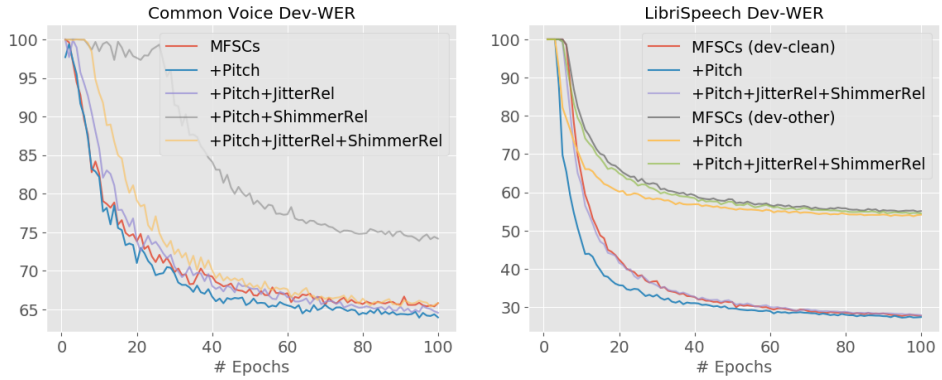


Figure 3.1: Common Voice and LibriSpeech dev set WER (%) during training, as a function of the epoch number. For the latter, dev-clean and dev-other are evaluated. Curves across the 5 different feature configurations for the same acoustic model architecture.

3.1.3. Conclusions

In this study, we performed a preliminary exploration on the effects of pitch and jitter/shimmer voice quality measurements within the framework of the ASR task performed by convolutional neural network models. The experiments reported with a publicly available Spanish speech corpus showed consistent improvements on the model robustness, achieving a reduced relative 7% WER in some scenarios. Besides, these feature extraction functionalities were provided and integrated with wav2letter code to easily replicate our findings or directly applying pitch and voice quality features to wav2letter models. We also provided the recipe for the Common Voice Spanish dataset, the first recipe suited for wav2letter using a Spanish publicly available dataset. The recipe for LibriSpeech experiments was also provided, which achieves up to a 2.94% relative WER improvement. Find both recipes in a GitHub repository[2].

Note that the approach employed for the feature combination was simple, by just appending such features to the spectral ones at the input layer, without extensive post-processing either nor adaptation of the model architecture. Being so, it was reasonable to think that there was still margin of improvement in the application of pitch and voice quality measurements to state-of-the-art neural models. Possible strategies comprised adapting the feature concatenation, maybe by dedicating exclusive filters to the new pitch and voice quality features. This was done in the following work, where we also switched from a convolutional to a Transformer-based architecture, showing better effectiveness.

[2]https://github.com/gcambara/wav2letter/tree/wav2letter_pitch

3.2. Pitch and Voice Quality Features for Transformer-based Speech Recognition

After obtaining improvements with pitch and voice quality features for a convolutional ASR model, we were encouraged to iterate the previous work with new additions. First of all, as Transformer models had been taking over the architecture landscape, we were motivated to replace our CNN model by one based on Transformers. Especially, as we hypothesized that the global attention mechanism of Transformers might work better with voice quality features, since these have a suprasegmental nature. Our intuition was that the incorporation of such features would assist the attention mechanisms, in order to disambiguate better the beginning and ending of words.

Furthermore, we evaluated the effectiveness of two methods for incorporating voice quality (VQ) and pitch features into the spectral coefficients, which are widely used in most neural automatic speech recognition (ASR) systems. The first method involves simply concatenating the VQ and pitch features with the spectral coefficients before the neural network's forward pass. The second method involves using two separate convolutional front-ends, one for spectral features and another for VQ and pitch features, and concatenating the resulting features from both front-ends. We used the LibriSpeech dataset (Panayotov et al., 2015) and the Transformer-based model (Vaswani et al., 2017) from fairseq's speech-to-text (S2T) recipe (Ott et al., 2019; Wang et al., 2020a). To the best of our knowledge, this was the first attempt to incorporate jitter and shimmer features into a modern Transformer-based speech recognition system while maintaining easily identifiable psychical/functional properties of the voice and linking them to ASR performance.

3.2.1. Model Description

To incorporate voice quality and pitch features into spectral coefficients, a Transformer-based model called S2T Transformer was used. The structure of the model comprises a front-end convolutional block, sinusoidal positional embedding of features, and an encoder-decoder that includes Transformer blocks, similar to (Vaswani et al., 2017). The convolutional block is composed of two 1-dim convolutional layers with a kernel size of 5, a stride of 2, and padding of 2, each of which uses GLU activation functions (Dauphin et al., 2017). The first layer accepts T time frames per $N = 40$ mel-spectrograms as input, upsamples them to 1024 features, which are then halved by the GLU activation function, producing $P = 512$ hidden features. These are passed to the second layer, which also outputs

512 features and is followed by GLU activation, resulting in $O = 256$ features, which is the embedding size used for all subsequent attention layers.

Adding voice quality and pitch features of size $M \in [1, 5]$ to the spectral features leads to an input feature vector of size $N + M$. As a result, the output feature vector of size $O = 256$ contains implicit pitch-related information. However, we were concerned that the information may become too diluted after convolution with the mel-spectrograms, since N is much greater than M.

To address this concern, we proposed an alternative architecture called S2T VQ Transformer, which is illustrated in Figure 3.2. This model comprises two convolutional blocks, A and B, which respectively receive mel-spectrograms and voice quality with pitch features. Both blocks perform independent convolutions and concatenate their outputs, resulting in a feature vector that represents K spectral features and L pitch-related features, giving a total output size of $O' = K + L$. By adjusting the balance between K and L, we can assign more weight to the new features, thereby enabling the attention layers to give them more attention. Block A and B are identical to the convolutional block in the S2T Transformer architecture, except for the modifications described in Table 3.3.

Table 3.3: Convolutional front-end blocks for mel-spectrograms and voice quality + pitch features.

Block	**Input Dim**	**Hidden Dim**	**Output Dim**
A	$N = 40$	$p_A = 512$	$K = 192$
B	$M \in [1, 5]$	$p_B = 256$	$L = 64$

It is important to note that the hidden dimension p_A is maintained for the spectral block A, but for block B it is halved, as there are fewer pitch and voice quality features and thus less upsampling is needed. We set the output dimensions for each front-end at $K = 192$ and $L = 64$, so that concatenation of these yields the same total output size as the plain S2T Transformer model, which is $O' = O = 256$. This ensures fairness in the comparison between the plain S2T and VQ S2T models, as simply increasing O' could lead to better results due to the increased number of parameters in the feature vectors, which could potentially be better exploited by the attention mechanism. It is worth noting that the VQ variant has slightly fewer parameters, with 29.2M compared to 29.4M in the plain model. Additionally, this design choice maintains a proportion of 1 pitch-related feature for every 3 spectral features, giving greater weight to the former compared to the plain S2T Transformer model.

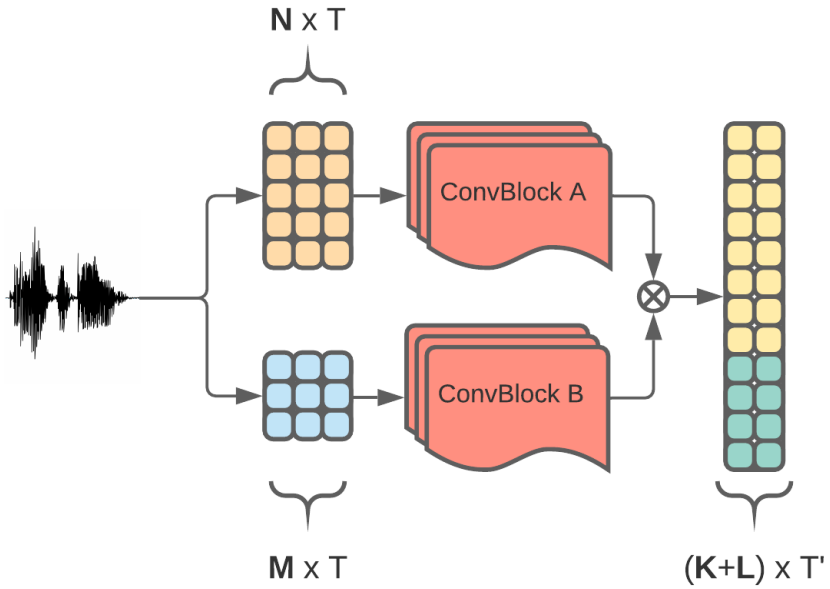


Figure 3.2: Convolutional front-end for mel-spectrogram filterbanks and pitch related features in S2T VQ Transformer architecture.

3.2.2. Methodology

Having introduced our model proposal, let's now examine our methodology, focusing on the dataset used, our experimental setup, and the choices made for acoustic modeling and decoding.

Dataset

This study evaluated its proposal using the LibriSpeech dataset (Panayotov et al., 2015), a collection of audio book recordings in English containing up to 1000 hours of speech that were sampled at 16kHz. The dataset has a pre-existing training split that was already divided into three subsets with around 100, 360, and 500 hours of speech, and two subsets were also created for the development and test sets. These subsets are referred to as train-clean-100, train-clean-360, train-other-500, dev-clean, dev-other, test-clean, and test-other.

Experimental setup

The main objective of this research was to investigate the significance of pitch and voice quality features in the speech recognition system. We trained and tested independent acoustic models with different feature combinations including mel-spectrograms, fundamental frequency (F0), POV, ΔF0, jitter and shimmer. We also tried a feature configuration with three random numbers sampled from a uniform distribution between 0 and 10, similar to mel-spectrogram filterbanks, to make sure that additional voice quality and pitch features do not yield better results just by increasing the model parameters. The performance of the model was evaluated in terms of WER in the test-clean and test-other sets, and the entire process was repeated for every feature configuration across 6 different initialization seeds to ensure statistical significance of the results. Due to computational limitations, the experiments were conducted only on the train-clean-100 set. Finally, we compared the performance of the filterbank-only baseline with the S2T VQ model, which includes pitch and voice quality features, with different training set hours: 50, 100, 200, 500 and 960.

Acoustic modeling

The process of training and testing the acoustic models was carried out using the fairseq toolkit (Ott et al., 2019). This toolkit provides examples and features that are useful in the speech-to-text task (Wang et al., 2020a). The LibriSpeech ASR example was used as a starting point for this experiment. The S2T small Transformer model, which has 31M parameters, was trained with mel-spectrogram features that were computed on-the-fly by fairseq.

To perform experiments that require pitch and voice quality features, Praat-Parselmouth (Jadoul et al., 2018), which is a Python wrapper for Praat (Boersma and Van Heuven, 2001), was used to precompute such features. The window size and stride for computing all the features were set to 25 ms and 10 ms, respectively. As the pitch, jitter and shimmer extraction algorithms only produce values for voiced frames, empty values for unvoiced frames were interpolated with the adjacent non-empty values in accordance with Kaldi's pitch extraction algorithm recommendation (Ghahremani et al., 2014). The POV vector, which indicates voiced (1) and unvoiced (-1) segments, was constructed by keeping empty indexes. The logarithm was applied to the pitch vector, and the pitch, jitter and shimmer vectors were smoothed out by subtracting the mean of a window of 151 frames, centered in the current frame, similar to Kaldi. Finally, cepstral mean and variance normalization was used to normalize all the combined features.

We tackled the treatment of features in two different ways, depending on the

selected front-end, as described in section 3.2.1: either by merely concatenating them with mel-spectrograms, or by processing them in a dedicated convolutional block and then concatenating them with the convolved mel-spectrograms. We utilized 40 filterbanks instead of the original 80 filterbanks in the fairseq example to increase the proportion of pitch and voice quality features in the overall number of features. This was done to enhance the numerical importance of the new feature set, specifically for the simple concatenation experiments, while still maintaining a commonly used number of filterbanks. As the simple concatenation configuration was not used for the training set size experiments, we reverted to 80 filterbanks.

The models were trained using the cross-entropy loss with label smoothing and Adam optimization. To prevent gradient explosions, gradient values above 10.0 were clipped. The learning rate was warmed up for 10k batch updates and then peaked at 0.002, and subsequently decreased by an inverse root scheduler. For the experiments involving different training sizes, the same hyperparameters as the fairseq recipe were used to ensure the baseline was closely matched. The models were trained until validation loss plateaued for many iterations to guarantee convergence. The 960 hours model was trained for 300k updates, the 500, 200, and 100 hours models for 150k updates, and the 50 hours model for 55k updates. However, for the feature configuration scan across seeds, the number of iterations was limited to 20k batch updates to save computational resources since longer training did not lead to significant improvements.

Decoding

We concluded that averaging the weights from the last 10 checkpoints during training yielded better WER scores than choosing the checkpoint with the best development WER, which was consistent with the method used in the fairseq recipe. Decoding was performed using the beam search algorithm with a beam size of 5 and a 10k unigram lexicon created with sentencepiece (Kudo and Richardson, 2021) from the LibriSpeech corpus. The purpose of this research was to evaluate the impact of voice quality and pitch features on acoustic modeling, so no additional language model was employed.

3.2.3. Results

Figure 3.3 shows the distributions of WER scores for a range of feature combinations with the six different seeds used on the LibriSpeech test sets. On the whole, the mean WER scores for the S2T VQ model are generally lower than

those for the plain S2T Transformer model using simple feature concatenation. Additionally, there is a clear trend towards lower WER scores as the number of added features increases, particularly for the S2T VQ model. This is particularly noteworthy when compared to the baseline of 40 filterbanks or the model that uses three randomly selected features.

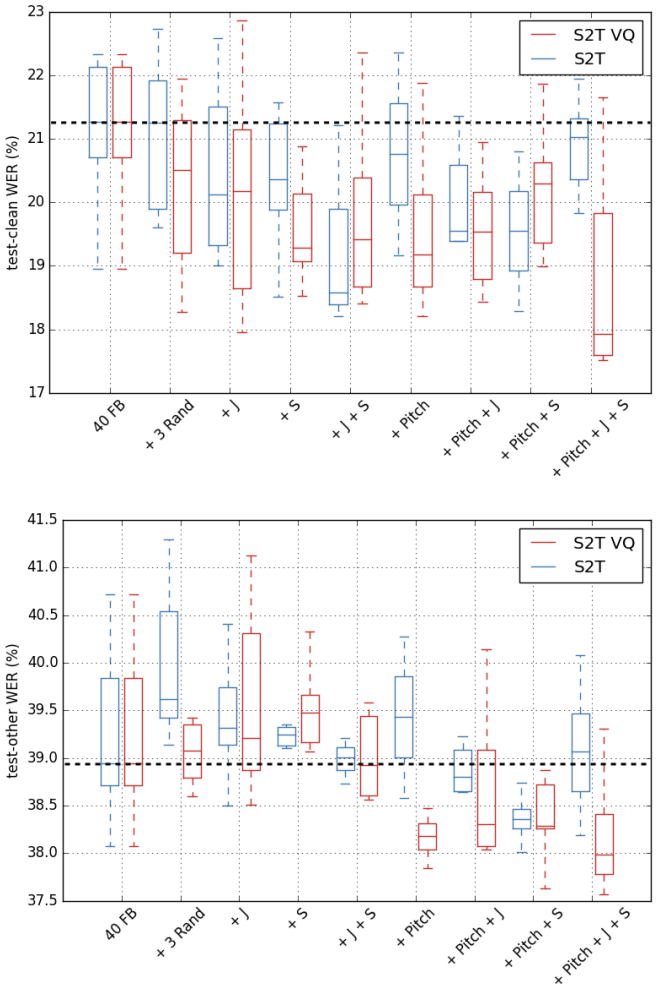


Figure 3.3: LibriSpeech test-clean and test-other WER (%) for several feature configurations tested, combining mel-spectrogram filterbanks (FB), 3 random features (Rand), F0+POV+ΔF0 (Pitch), jitter (J) and shimmer (S), using S2T and S2T VQ models. 40 FB baseline is marked by the dashline.

The results show that using specific convolutions for pitch and VQ features as hypothesized is beneficial. While jitter and shimmer have only a slight positive impact on the performance of the system, improving mean WER by 1.3%

and 0.2% for test-clean and test-other respectively, pitch achieves a better performance, with an improvement of 1.5% and 1.0% mean WER for the same sets. However, combining pitch with jitter and shimmer results in the largest improvement, with a mean WER reduction of around 2.3% for test-clean and 1.0% for test-other. The pitch experiments show fair significance, with two-tailed p-values of 0.134 and 0.069, while the jitter and shimmer experiments show low significance, with p-values of 0.129 and 0.624. However, the experiments with pitch and VQ features combined show the most significant results, with p-values of 0.027 and 0.052 for test-clean and test-other. This suggests that jitter and shimmer may be a good complement to improve pitch features, but alone, they do not cause a significant change.

The data in Table 3.4 indicate that the S2T VQ model with pitch and VQ features outperforms the baseline filterbank-only model in most of the training size scenarios. On average, the WER reduction is 5.6% for test-clean and 3.0% for test-other, showing that the advantages of using pitch and VQ features hold even as the amount of training data increases. To gain a better understanding of how pitch and VQ features affect performance as the amount of training data grows, we can examine the distribution of error types shown in Figure 3.4. The results indicate that while the evolution of deletions (D) is similar in both models, the S2T VQ model with pitch and VQ features significantly reduces the number of insertions (I), resulting in a higher proportion of substitutions (S). This suggests that with sufficient data, the S2T VQ model can learn to better use prosody information to distinguish the beginning and end of words.

Table 3.4: Mean test WER scores for acoustic models trained with different subset sizes of LibriSpeech training. Two architectures are used: the baseline with filterbank features (80 FB) and the S2T VQ model with filterbank, pitch and voice quality features (+VQ).

Train set	WER (%)			
Hours	**test-clean**		**test-other**	
	80 FB	**+ VQ**	**80 FB**	**+ VQ**
50	29.26	**28.30**	47.70	**46.91**
100	19.33	**18.29**	37.04	**35.31**
200	10.95	**10.80**	**24.94**	25.32
500	7.78	**6.96**	18.34	**17.56**
960	4.62	**4.27**	10.64	**10.01**

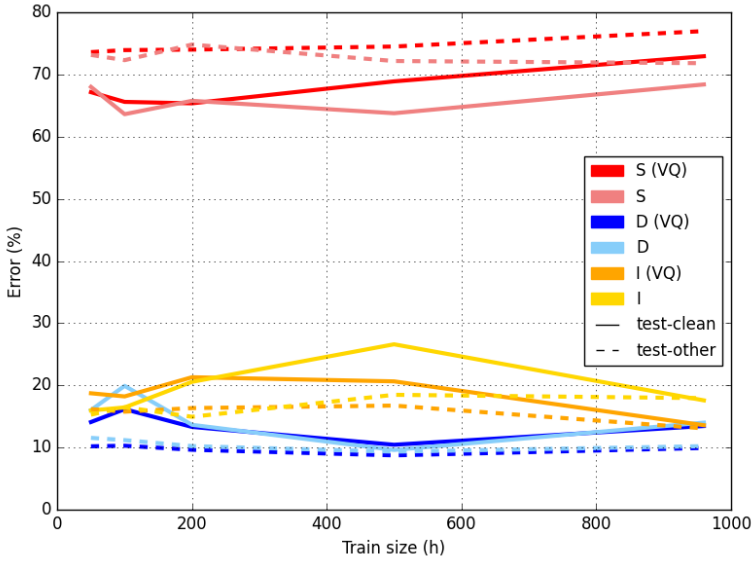


Figure 3.4: Error distributions across training hours. Error types: substitutions (S), deletions (D) and insertions (I).

3.2.4. Conclusions

This study investigated the impact of incorporating pitch and voice quality features into the spectral features of a state-of-the-art acoustic ASR using two methods: simple concatenation and separate convolutional filters. The latter approach outperformed the former, resulting in significant WER improvements when using pitch and voice quality features separately and when combined, with relative WER reductions of up to 5.6%. Using separate convolutional filters for these features increases the representation of prosodic information in the acoustic features, which can be better leveraged by the attention layers. The results suggest the potential of voice quality features for complementing prosodic information carried in pitch features, and motivate further research in conversational speech recognition or speech recognition with punctuation marks. We also suggest exploring other prosodic features such as intensity and rhythm beyond pitch. Lastly, we made our recipe available to the research community in a GitHub repository [3].

[3] https://github.com/gcambara/speechbook/tree/master/recipes/vq_pitch

Chapter 4

IMPROVING RECOGNITION IN NOISY ENVIRONMENTS WITH SPEECH ENHANCEMENT

Speech recognition is a complex task, as mapping speech to text means abstracting to a lot of low-level details in audio to obtain a precise transcription. In Chapter 3, we saw how pitch and voice quality features induce additional information that help ASR models to disentangle prosodic information to better delimit beginning and ending of words. However, audio recordings do not only contain speech signal, which is already difficult to transcribe, but also register background noises that add further complexity to the transcription task. Many day-to-day recordings contain noises from cars, wind, background conversations or ringing phones, so how do we minimize the impact of these in the automatic transcription process?

This is the question that we address in this chapter of the thesis. Particularly, we focus on a technique called speech enhancement (SE), which consists in cleaning out the background noises from an audio signal, whether it is at the waveform, spectral or latent level. This method is not only used for ASR, and actually it is more commonly found in applications where perceptual quality is important. We can think of videocall or audio editing applications, where we may want to remove undesired noises, in order to hear speech better. There are classic SE methods like Wiener filtering (Meyer and Simmer, 1997), spectral subtraction (Yang and Fu, 2005) or subspace algorithms (Ephraim and Van Trees, 1995), but modern neural architectures are providing better results, especially when dealing with non-stationary noises or overlapping speech. As mentioned, SE can be used for better perceptual quality, but its usage can benefit also the purpose of ASR, by

obtaining features with less representation of noise, which would hypothetically yield better transcription results for a neural classifier. We were able to study this for two different cases with noisy conditions: firstly, transcribing speech from TV shows, and secondly, detecting a wake-up word for a real AI home assistant.

IberSPEECH is a conference for the speech community in the Iberian Peninsula, which has been helding the Albayzin challenge during the last years. One of the proposed tasks in this challenge is the recognition of speech from Spanish TV shows. This case is very interesting for ASR research, not only because of the expressiveness and variety of speech, but also because audio extracted from TV shows has complex acoustic conditions. TV shows happen in a wide variety of setups, whether they are indoor or outdoor, with several types of noises coming from different sources, plus background music that is commonly used. As we participated in the challenge, we did not only look to achieve the best WER scores to win it, but also committed some computational budget to explore SE techniques to clean the audio, and explored its impact on the transcription process. From these experiments, detailed in the following sections, we concluded that using out-of-the-box SE modules was a promising technique, but it also had its drawbacks. SE models helped to transcribe speech better if the audio was severely contaminated with noise, but if the audio was already clean enough, SE introduced additional acoustic artifacts that made transcriptions less precise. Such results would encourage us to conduct deeper research on SE applied for speech recognition, after the Albayzin challenge. This time, we would investigate it in a different scenario: recognizing the words used for waking up a home AI assistant, called Aura.

Aura is the home assistant of the Telefónica company. It provides many services, like informing about the weather, TV show schedules or playing movies for the customer. All of this is done through voice interaction, so Aura's device remains idle at home, until the user says "OK Aura", which are the wake-up words to trigger the conversation. People interact with home assistants with home noises happening in the background, like open faucets, dogs barking or TV shows on. With the aim of improving Aura's wake-up word module, and motivated by the results of the Albayzin challenge, we conducted research on the application of SE for Aura. This time, the speech recognition scope was narrowed down by definition, as we only had to recognize the wake-up word, instead of a large vocabulary. For such a reason, effective wake-up word modules tend to be smaller than large vocabulary ASR ones, so we would be able to commit more computational resources on exploring the SE part. As a consequence, we explored several ways of applying SE for better speech recognition: from applying separately trained models, to jointly training SE and wake-up word detection models to optimize the latter task. From such explorations we found clearer patterns of how SE affects the recognition task, for a variety of signal-to-noise ratios (SNR). Our results

suggest that joint end-to-end training of SE and wake-up word models bring the most robust behavior, for any SNR range. We detail all our findings, within the Albayzin challenge and the Aura project, in this chapter.

4.1. Speech Enhancement for Speech Recognition in TV Shows

The Albayzin 2020 challenge consisted of several tasks, one of these being performing ASR on TV shows from the Spanish National TV (RTVE). We participated in the ASR challenge, in a joint effort between the Brno University of Technology (BUT), Telefónica Research (TID) and Universitat Pompeu Fabra (UPF), presenting a variety of systems from hybrid to end-to-end neural models (Kocour et al., 2021). Specifically, we submitted three different systems: a hybrid model (Povey et al., 2018), a neural end-to-end Conv GLU model (Collobert et al., 2016) (very similar to the one used in Section 3.1) and a joint fusion of both. The fusion consisted of a word-level ROVER (Fiscus, 1997) fusion, and it achieved a 23.33 % WER on the official evaluation dataset, the 4th best score in the challenge.

Furthermore, as previously mentioned, we used specific models to filter out noises from TV shows, and used the cleaned signal to evaluate our convolutional end-to-end system. This is the main part that we describe in this section, as we are focused on the usage of SE models to obtain features that are free of noise, or at least, as much as possible. Particularly, we employed two types of models to do this: a speech denoiser, called Denoiser (Défossez et al., 2020), and a music source separation model, called Demucs (Défossez et al., 2019). The intuition for the latter model was that as it separates vocal tracks from the rest of instruments, it could be used for extracting speech from background noises and music.

4.1.1. Methodology

Let's delve into the methodology employed for evaluating SE in transcribing TV shows. This includes a detailed description of the datasets used in our study, the end-to-end convolutional ASR model utilized, and specifics of the training and decoding processes.

Data

The Albayzin 2020 challenge offered two databases, namely RTVE2018 and RTVE2020. RTVE2018 was used for training and development, while RTVE2020

was used for the final evaluation of the submitted systems. On the one hand, RTVE2018 database (Lleida et al., 2018; Lleida et al., 2019) includes 15 TV programs aired by Radiotelevisión Española (RTVE) and offers various speech scenarios, such as read speech, spontaneous speech, live broadcast, and political debates. This database has a total of 569 hours of audio data, out of which 468 hours come with subtitles and are assigned as train set, and the remaining 109 hours are human-revised and divided into dev1, dev2, and test sets. Both hybrid and end-to-end models utilized dev1 and train sets for training, while dev2 and test sets were used for validation purposes. On the other hand, RTVE2020 database (Lleida et al., 2020) consists of 70 hours from TV shows that are manually annotated and broadcast by RTVE.

The end-to-end system was trained with additional data from multiple sources, including the Fisher Spanish Speech dataset (Graff et al., 2010), Mozilla's Common Voice Spanish corpus (Ardila et al., 2020), and Telefónica's Call Center inhouse data, which is 23 hours long. Mozilla's Common Voice Spanish corpus is a publicly available dataset consisting of recordings of people reading scripted sentences, with a sampling rate of 48kHz. The sentences were donated by volunteers and taken from public domain movie scripts. The version of the Common Voice corpus used in this study was 5.1, which includes 521 hours of recorded speech. However, only the speech that was validated by the contributors, which amounts to 290 hours, was used.

The RTVE2018 database used for training contains a large amount of speech that has been subtitled. However, these captions contain numerous mistakes. Often, the captions are out of synchronization by a few seconds, meaning that the correct transcript corresponds to a different part of the audio. This problem also affects the human-revised development and test sets. Additionally, there are "partly-said" captions, which include misspelled and unspoken words in the transcription.

To avoid errors in the training process of the hybrid ASR system due to misaligned labels, a transcript retrieval technique developed in (Manohar et al., 2017) was used. The technique involves concatenating closed captions related to the same audio in their original timeline, creating a small text corpus of a few hundred words. This text corpus is used to train a biased N-gram language model with $N = 7$, so that the model is only biased towards the currently processed captions. During decoding, the weight of the acoustic model is less than the weight of the language model since it is believed that the captions should appear in hypotheses. The "winning" path is then retrieved from the hypothesis lattice as the path with the minimum edit cost with respect to the original transcript. Finally, the retrieved transcripts are segmented using the Continuous Time Marked (CTMs) files obtained from the oracle alignment, and any segments that do not correspond

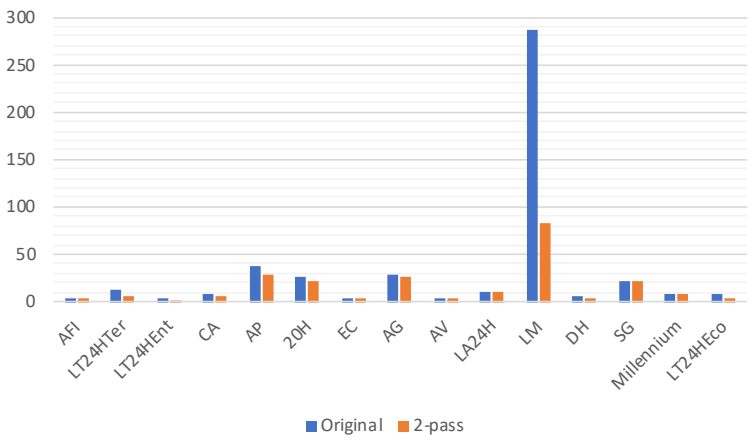
to the original transcripts are discarded. More information on this technique can be found in (Kocour, 2019) and the paper by (Manohar et al., 2017), from which it was adapted.

The transcript retrieval technique was used twice in the process. Firstly, an initial ASR model was trained on out-of-domain data like Fisher and CALLHOME (Canavan and Zipperlen, 1996). Such model was used to generate hypotheses in the first pass of transcript retrieval. Next, a new system was trained from scratch using the clean data from the first pass, and the entire process of transcript retrieval was repeated. Table 4.1 demonstrates that this two-pass cleaning approach results in the retrieval of nearly all the manually annotated development data and 50% of the subtitled training data.

Table 4.1: Two-pass transcript retrieval.

Cleaning	Train	Dev1	Dev2	Test
Original	468	60.6	15.2	36.8
1-pass	99.4	21	7.5	-
2-pass	234.2	55.1	14.3	33.7
Recovered	50 %	91 %	94 %	92 %

Figure 4.1: Amount of cleaned audio per TV-show, in hours.



The barplot displayed in Figure 4.1 illustrates the number of hours that were recuperated for different TV programs. Additionally, it provides insights into how the data was spread out in the database. The majority of the speech data was sourced from the La Mañana (LM) TV program. However, due to its complexity, our ASR model struggled to accurately transcribe the audio data from this

program, and as a result, we had to discard most of the data after conducting a thorough two-pass data cleaning process.

End-to-end Convolutional Speech Recognition

The acoustic model we used is an end-to-end design that employs the convolutional architecture developed in (Collobert et al., 2016) and incorporates gated linear units (GLUs). GLUs are utilized in convolutional models to prevent the issue of vanishing gradients and maintain high performance by offering linear pathways. Specifically, we utilized the model from the wav2letter Wall Street Journal (WSJ) recipe, which has around 17 million parameters and applies dropout after each of its 17 layers. The WSJ dataset has about 80 hours of audio recordings, which is smaller than our dataset (around 600 hours). Although the LibriSpeech recipe utilizes a deeper Conv GLU-based architecture with around 1000 hours of data, we opted for the WSJ recipe to reduce computation time and enhance fine-tuning of the network's hyper-parameters.

To train the system, the wav2letter framework was used and all data samples were resampled to 16 kHz. Mel-frequency spectral coefficients (MFSCs) were extracted from the raw audio using 80 filterbanks, and the Auto Segmentation criterion (ASG) was used with a batch size of 4 (Collobert et al., 2016). The learning rate began at 5.6 and was gradually reduced to 0.4 after 30 epochs, at which point the training was stopped since no significant Word Error Rate (WER) improvements were observed. From epochs 22 to 28, the system was trained using the same training set, but with the addition of RTVE2018 train and dev1 samples that had background music cleaned by the Demucs module (Défossez et al., 2019). In the last two epochs (28-30), further samples were included that had background noise removed by Demucs and furtherly denoised by a Denoiser (Défossez et al., 2020). This was done to augment the data with samples that had challenging acoustic conditions, thus helping the network to learn to generalize audio artifacts caused by the denoiser and music separator networks, which would be beneficial when using these networks to clean test audio.

The lexicon used in this study was created from the train and validation transcripts, along with the Sala lexicon (Moreno et al., 2002). It was a grapheme-based lexicon containing 271,000 words, with 37 tokens including the standard Spanish alphabet, the "ç" letter from certain Catalan words, and vowels with diacritical marks. The language model (LM) used in this study was a 5-gram model trained with KenLM (Heafield, 2011), using only transcripts from the training sets RTVE2018 train and dev1, Common Voice, Fisher, and Call Center. The resulting LM is named in this chapter as Alb+Others.

Decoder hyperparameters were fine-tuned using the RTVE2018 dev2 set, and the best results were achieved with an LM weight of 2.25, a word score of 2.25, and a silence score of -0.35. This same configuration was used for evaluating the RTVE2018 and RTVE2020 datasets.

4.1.2. Speech Enhancement Experiments

Background music is frequently present in TV programs and can cause confusion for speech recognition systems if it is prominent. To address this issue, we used a Music Source Separator called Demucs (Défossez et al., 2019) to process the audio. Demucs separates the original audio into different components such as voice, bass, drums, and others. By keeping only the voice component, we were able to reduce background music while keeping good quality in the speech signal.

We attempted to perform speech enhancement on the validation sets to see whether if we could lower the WER. However, the results indicated that this approach only led to a small increase in WER, as shown in Table 4.2. We also experimented with using a specialized denoiser (Défossez et al., 2020) after removing the background music. Unfortunately, this approach increased the WER for dev2 by 1.6% compared to the original system without any enhancement. Initially, neither of these two approaches (Demucs and Demucs + Denoiser) improved the WER, so we did not use them for the end-to-end model in the final fusion. However, the UPF-TID team submitted separate systems using the end-to-end, end-to-end + Demucs, and end-to-end + Demucs + Denoiser models, which are listed in Table 4.3. Post-eval results are obtained after bug-fixing of the system used in the official submission. Notice that performance in RTVE2020 dropped around a 15% WER, compared to RTVE2018 dataset. Our main guess is that probably the model overfitted to the acoustic conditions and voices of the TV shows present in RTVE2018, thus struggling with new shows in RTVE2020.

Table 4.2: Overall results on RTVE2018 dataset depending on the usage of language models and speech enhancement.

	AM	LM	WER [%] Dev2	WER [%] Test
1	Conv GLU	None	36.1	37.5
2		Alb + Others	**20.8**	**20.7**
3	+ Demucs	None	36.4	37.5
4		Alb + Others	21.1	20.8

Table 4.3: Official and post-evaluation final results on RTVE2020 eval set for the end-to-end systems.

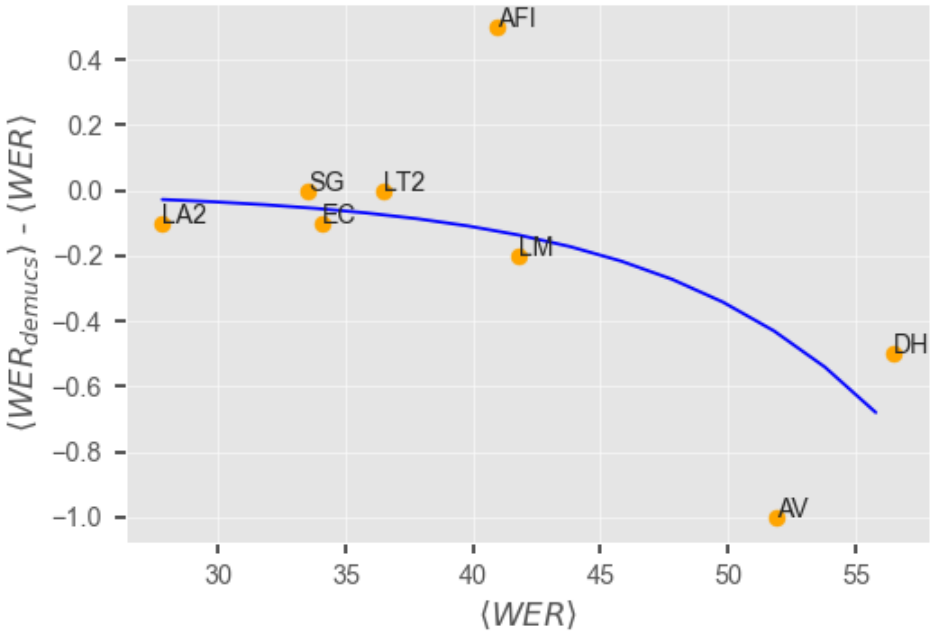
Model	WER [%]	
	Official	Post-eval
Conv GLU	**41.4**	**36.2**
+ Demucs	42.3	37.9
+ Demucs + Denoiser	58.6	40.0

Our hypothesis for the drop in transcription quality is related to the fact that many samples do not have background music. If we enhance the speech in these clean samples, it may actually harm the quality of the signal. To test this, we applied music source separation to samples with different SNR ranges, measured by the WADA-SNR algorithm (Kim and Stern, 2008). Our evaluation of the RTVE dataset shows that music separation is most effective for SNR ranges between -5 and 5 or 8, as demonstrated in Table 4.4. The greatest improvements were seen in TV shows with higher WER, indicating more difficult or noisier speech, such as AV, where the speakers are often in a car, or LM and DH, where music and speech frequently overlap. Other shows already had good quality audio, so the benefits were not as significant. However, the AFI show had poor quality audio, and applying Demucs could potentially make it even worse, resulting in decreased performance.

Table 4.4: WER impact of cleaning speech signals between certain SNR ranges, using a music source separator. End-to-end Conv GLU model is used without LM, and percentage of cleaned samples are reported.

SNR	Cleaned Samples [%]		Test WER [%]	
	2018	2020	2018	2020
$(-\infty, \infty)$	100	100	37.50	53.53
$(-\infty, 10)$	25.97	34.22	-0.05	-0.87
$(-5, 10)$	25.84	31.33	-0.05	-0.88
$(-5, 5)$	5.14	11.88	-0.07	**-1.03**
$(-5, 8)$	14.95	22.11	**-0.08**	-0.97

Figure 4.2: Variation of the mean WER per TV show between using Demucs-cleaned or original samples on RTVE's 2018 test set. Negative values represent Demucs improvements. Note that only samples with SNR between -5 and 8 are enhanced.



4.1.3. Conclusions

By the end of the Albayzin 2020 Challenge we had applied enhancement techniques to denoise the highly complex audio from TV shows data. The main finding was that, despite effectively cleaning noise from audio, the models used for enhancement (Demucs and Denoiser) introduced additional artifacts to samples that were already clean. This made the mean WER a bit higher, an undesired effect. After inspecting WER reductions for different SNR intervals, we found out that cleaning speech was delivering better WERs when applied on noisy audio. Having used this out-of-the-box denoiser and source separation models, we concluded that SE for ASR would be better done in a tailored manner. This would imply triggering SE models at low SNR ranges, or training the speech recognition models to be agnostic of artifacts introduced by SE modules. The latter idea was furtherly explored in our following work: applying SE for the wake-up word detection in a real-world voice assistant, Aura.

4.2. Task-Aware Speech Enhancement for Wake-up Word Detection

Detecting wake-up words in noisy environments is crucial for ensuring a positive user experience with voice assistants. The problem is that the device can be activated unintentionally due to background noise from conversations, TVs, or other household devices. With such problem in mind, we decided to tackle wake-up word detection with previous enhancement of the speech signal, as we did during the Albayzin 2020 Challenge, described in Section 4.1. Taking into account the past experience in the challenge, where the standalone enhancement model would introduce audio artifacts in clean speech, this time we decided to treat both speech enhancement and recognition neural models in a joint manner. With that in mind, we hypothesized that the speech enhancement would clean the audio in order to prevent errors in the recognition side.

Our new study proposed a solution to improve wake-up word detection in noisy environments by using a speech enhancement convolutional autoencoder combined with on-device keyword spotting. The system is trained end-to-end, optimizing a combination of losses, including a reconstruction-based loss at both the log-mel spectrogram and waveform levels, as well as a task-specific loss that considers cross-entropy error for keyword spotting detection. The study experimented with various neural network classifiers and found that coupling speech enhancement with wake-up word detection significantly improves performance in noisy conditions, particularly when the two are jointly trained. Additionally, a new publicly available speech database for the Telefónica's voice assistant, Aura, called the OK Aura Wake-up Word Dataset (Cámbara et al., 2022), was introduced. The dataset includes rich metadata, such as speaker demographics and room conditions, and contains carefully selected hard negative examples that exhibit different levels of phonetic similarity to the trigger words "OK Aura."

Let's get into the details of this research's context, which work was done for Aura, a cognitive conversation system. These systems rely on speech as the most natural means of communication. A crucial component of these systems is the speech-to-text (S2T) technology that accurately transcribes the user's speech into text for further processing in the natural language engine. However, to avoid running S2T models in the background at a high computational cost, a wake-up word (WUW) is often required to trigger the S2T functionality and other conversational mechanisms. The WUW module is designed to distinguish only between the trigger word and other acoustic input, making it a two-class hypothesis test that is less computationally and resource-intensive than an always-awake S2T model.

Although the WUW model is simpler than a large vocabulary automatic speech

recognizer, it still needs to be resilient enough to handle acoustic disturbances like TV, music, or background conversations. In noisy environments, the WUW's performance is affected by unexpected wake-ups, resulting in false alarm errors, and failing to detect the trigger word, known as miss errors. These errors, particularly false alarms, have a significant impact on user experience and can reduce expectations and engagement with the technology. For this reason, there a few common approaches used to improve the robustness of WUW detection. One such approach involves a second-step verification process, which typically involves either an ASR or a WUW model (Ge and Yan, 2017; Kumar et al., 2020; Michaely et al., 2017; Apple, 2017). Other works incorporate a SE module that operates at the audio input stage to reduce noise and obtain a cleaner version of the acoustic signal. The SE module aims to enhance the perceptual quality and intelligibility of speech by removing background noises (Loizou, 2013; Xu et al., 2013). While Speech Enhancement (SE) is commonly used in telecommunications and hearing aids to improve perceptual experience, it has also shown to improve results when used as a pre-processing step in the context of ASR task (Zorilă et al., 2019; Maas et al., 2012; Weninger et al., 2015).

Looking at WUW task particularly, recent studies have introduced systems based on neural architectures, such as convolutional (Sainath and Parada, 2015), recurrent (Kumar et al., 2018; Arik et al., 2017; Yamamoto et al., 2019), and self-attention networks (Shan et al., 2018). To address robustness and generalization issues, a commonly used strategy is to generate training data by adding noise. This approach exploits the ability of the deep neural networks to handle large amounts of data by artificially corrupting the original samples. By doing so, the resulting models become more robust and can handle a wider variety of noises and scenarios. Similar techniques have been applied to various speech-related tasks, including keyword spotting (Raju et al., 2018), automatic speech recognition (Hsiao et al., 2015; Hannun et al., 2014), and wake-up word detection (Yoon and Kim, 2020). In this work, we use similar techniques to augment our training data for all the classifiers we describe. We add noise or create artifacts in the original speech to generate new training samples, which leads to improved performance in wake-up word detection tasks, consistent with findings reported in previous works on other speech tasks.

Classic SE methods like Wiener filtering (Meyer and Simmer, 1997) or spectral subtraction (Yang and Fu, 2005) are good at reducing noise from speech signals but not robust against certain types of noise, like non-stationary ones or overlapped speech. Deep learning approaches like encoder-decoder autoencoders have been proposed to address this issue. Popular models include those in (Pascual et al., 2017), using generative adversarial networks (GAN) (Goodfellow et al., 2014), or (Défossez et al., 2020), which acts at the waveform level in real time.

Many current approaches are optimized by minimizing a regression loss in time or a combination with a spectrogram domain loss (Park and Lee, 2016; Défossez et al., 2020). Inspired by these works, and our previous research on SE for TV shows transcription, we hypothesize that using SE to clean noisy speech can improve WUW detection. To test this hypothesis, we conduct various experiments on different model proposals:

(a) The isolated classifier: a baseline scenario where only a WUW classifier is used without any SE module.

(b) The independent SE and WUW models: these two systems are trained separately, so the SE model is optimized only waveform and spectral reconstruction losses.

(c) The Task-Aware SE (TASE) through frozen WUW model: the WUW model is trained first and then incorporated into the SE model during training. This allows the WUW detection logits to be used in the SE model as a classification loss, which is back-propagated along with the regression losses. The WUW detector is not updated during SE training.

(d) The end-to-end TASE (TASE-E2E) and WUW model training: it involves training both the SE and WUW models together from scratch using joint regression and classification losses.

Summarizing, we explore the application of neural SE to WUW detection, which had not been studied before, as far as we know. We introduce a new task-aware loss function that enhances speech for better performance in WUW detection. This is achieved by back-propagating both the regression loss from the SE module and the classification loss from the WUW classifier, as can be seen in Figure 4.3. We evaluate the performance of our approach under different SNR ratios and acoustic scenarios, demonstrating that SE is particularly effective in high-noise environments.

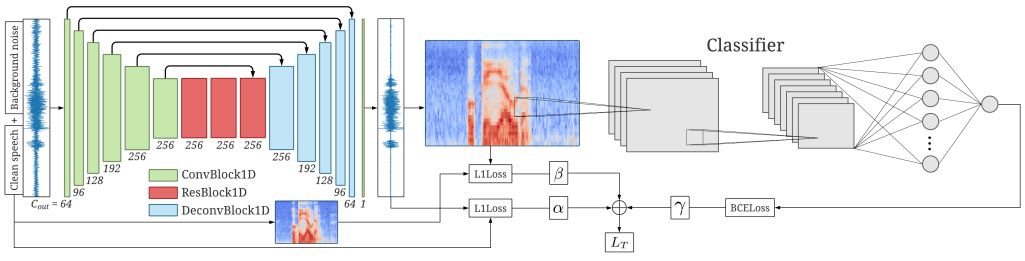


Figure 4.3: End-to-end TASE model at waveform level concatenated with a classifier.

4.2.1. Model Description

Our model is a type of SE model that uses a fully-convolutional denoising autoencoder with skip connections (as shown in Figure 4.3), which is similar to other successful SE models like (Pascual et al., 2017; Défossez et al., 2020; Llombart et al., 2021). During training, we give the model a noisy audio waveform $\boldsymbol{x} \in \mathbb{R}^T$, which consists of both a clean speech signal $\boldsymbol{y} \in \mathbb{R}^T$ and background noise $\boldsymbol{n} \in \mathbb{R}^T$. The model is trained to adjust the value of $\boldsymbol{x}$ such that it equals $\lambda \boldsymbol{y} + (1 - \lambda)\boldsymbol{n}$, where λ is a parameter that controls the SNR.

The encoder consists of six ConvBlock1D, which are sequences of a convolutional layer, instance normalization, and ReLU. In the ConvBlock1D, a kernel size of $K = 4$ and a stride of $S = 2$ are used except for the first layer, which uses $K = 7$ and $S = 1$. The compressed signal then goes through three intermediate ResBlock1D, which preserve the shape, and each block consists of two ConvBlock1D with $K = 3$ and $S = 1$. Skip connections are added from the input of each residual block to its output. The signal then flows through the decoder, which has the opposite structure of the encoder, and consists of DeconvBlock1D that replace the convolutional layers in the ConvBlock1D with transposed convolutional layers. The decoder produces the enhanced signal with the same shape as the input waveform, which is passed on to the WUW classifier. Both the encoder and decoder blocks are connected with skip connections to maintain low-level details of the waveform.

The model is fully convolutional to minimize the delay compared to an RNN-based architecture for the same task. Table 4.5 shows a comparison of our model with state-of-the-art architectures in terms of parameters, operations, size, and forward delay. To ensure fair comparison, we measured the forward time of each model using the same CPU and the same input data, an audio of 1.5 seconds. We ran 100 forward passes and calculated the average forward time. We also evaluated a variant of our architecture called "gruse" in which we replaced the residual blocks with a Gated Recurrent Unit (GRU) with a hidden size of 256. This resulted in a smaller model with fewer operations, but with a considerably higher forward delay. The other architectures listed in the table are demucs (H = 64 and H = 48) from (Défossez et al., 2020), and NSNet2, which is the baseline network used for the Deep Noise Suppression Challenge (Braun and Tashev, 2020).

Architecture	Parameters	# Operations	Size (MB)	Fwd Time (ms)
demucs (H = 64)	33.53M	10,015M	278.61	163.21
demucs (H = 48)	18.86M	5645M	184.05	98.28
NSNet2	2.80M	-	-	22.00
TASE	2.45M	4156M	154.67	65.50
gruse	1.31M	1853M	64.11	176.42

Table 4.5: Parameters, number of operations (multiplications and additions), size, and forward time of SE models.

We optimize the model with a combination of the L1 loss for the target raw waveform and the log-mel spectrogram, as proposed in (Yamamoto et al., 2020). With this, we aim to reconstruct the clean waveform $\hat{\boldsymbol{y}}$, but we also include a binary cross-entropy (BCE) loss from the WUW classifier, for the TASE case. There are two options when the BCE loss is used, either the WUW classifier is jointly trained from scratch with the SE module, or we use a pretrained WUW model, which we freeze. Using this BCE loss is supposed to help the TASE model to optimize audio cleaning towards WUW detection. Formally, we define the final loss function as a linear combination of three losses:

$$L_T = \alpha L_{raw}(\boldsymbol{y}, \hat{\boldsymbol{y}}) + \beta L_{spec}(S(\boldsymbol{y}), S(\hat{\boldsymbol{y}})) + \gamma L_{BCE}, \tag{4.1}$$

where α, β, and γ are the loss weights, and $S(\cdot)$ is the log-mel spectrogram of the signal, which is computed using 512 FFT bins, a window of 20 ms with 10 ms of shift, and 40 filters in the mel scale.

4.2.2. Methodology

In this section, we outline the methodology of our assessment, detailing the databases and data augmentation techniques used, as well as the wake-up word detection models evaluated, including training and testing nuances.

Databases

We gathered samples containing the WUW, "OK Aura", from two in-house databases from Telefónica. One of them is publicly available for research purposes (Cámbara et al., 2022), if requested through an End-User License Agreement (EULA). Additionally, we extracted more negative samples (without the

WUW), from the Spanish Common Voice (CV) corpus (Ardila et al., 2020). Furthermore, we got sounds for background noise contaminations from other datasets like QUT-NOISE (Dean et al., 2010) or the IberSpeech-RTVE Challenge (Lleida et al., 2019). We made sure that no speaker or background noise were repeated across training, validation and testing splits, which ratio was 80-10-10. The total size of the dataset was of 50,737 non-WUW samples and 4651 samples

Regarding the data collection of WUW samples, this was done in two rounds. Firstly, we collected 4300 samples (2.8 h) from 360 speakers, along with office background noise recordings. Secondly, we recorded 1247 utterances (1.4 h) from 80 speakers. This second round was motivated by the fact that we wanted to outsource a public dataset, so this time we asked participants to sign a consent form. Also, we captured sentences that were hard for baseline WUW models to recognize. Mostly, sentences that are close phonetically to the WUW, but not the WUW itself. We divided the sentences in different levels of similarity to the WUW:

1. The WUW itself: *OK Aura.*

2. The WUW within a context sentence: *Perfecto, voy a mirar qué dan hoy. OK Aura.*

3. Contains "Aura": *Hay un aura de paz y tranquilidad.*

4. Contains "OK": *OK, a ver qué ponen en la tele.*

5. Contains similar word units to "Aura": *Hola Laura.*

6. Contains similar word units to "OK": *Prefiero el hockey al baloncesto.*

7. Contains similar word units to "OK Aura": *Porque Laura, ¿qué te pareció la película?*

However, not all the difficulty for recognizing the WUW lies in phonetic similarity. There are other factors that may harm performance, like biases in gender, age or accent, plus acoustic conditions related to distance to the microphone or the room size. Metadata regarding all these topics was registered as well, as can be seen in Table 4.6.

Metadata	Values
Age	20s, 30s, 40s, 50s, 60s...
Gender	Female, Male, Non-binary
Distance	Close, Two steps away
Room size	Small (0–10 m^2), Medium (10–20 m^2)
Prosody	Unknown, Neutral, Annoyed, Friendly
Accent	Andalusian, Andean-pacific, Castilian, Non-native...

Table 4.6: Metadata in the OK Aura Wake-up Word Dataset.

We acquired data using Jotform[1], a web-based form service. The data was published with the name "OK Aura Wake-up Word Dataset" (Cámbara et al., 2021), and is available to the public [2]. Audio files in the OK Aura Wake-up Word Dataset are sampled at a 16 kHz rate, and we used a Speech Activity Detection (SAD) model to keep chunks where speech occurs. We used a model from pyannote.audio (Bredin et al., 2020) that was trained with the AMI dataset (Carletta, 2007).

As for non-WUW samples, we selected a subset of 55 h for training, 7 h for development and 7 h for testing, from the 300 h of the Spanish CV dataset (Ardila et al., 2020). The criteria behind this selection was to keep a ratio of 10:1 between negative and positive samples, as suggested in the literature (Hou et al., 2020). With respect to background noises, we used different contaminations like music, from the Free Music Archive[3] or conversations from Podcasts in Spanish [4]. Find more information on the datasets used and noise types in Table 4.7.

Noise Type	Dataset
Living Room	QUT-NOISE (HOME-LIVINGB) (Dean et al., 2010)
TV	IberSpeech-RTVE Challenge (Lleida et al., 2019)
Music	Free Music Archive
Conversations	Podcasts in Spanish
Office	In-house OK Aura WUW Dataset

Table 4.7: Background noise datasets.

[1] https://form.jotform.com/201694606537056
[2] https://zenodo.org/record/5734340
[3] https://freemusicarchive.org/
[4] https://www.podcastsinspanish.org/

Data augmentation

Music and TV original recordings were convolved with diverse Room Impulse Responses (RIR) based on the Image Source Method (Allen and Berkley, 1979), in order to simulate the reverberation of different room sizes (L_x, L_y, L_z), where $2 \leq L_x \leq 4.5, 2 \leq L_y \leq 5.5, 2.5 \leq L_z \leq 4$ m, with microphone and source randomly located at any (x, y) point within a height of $0.5 \leq z \leq 2$ m.

After testing different data augmentation techniques, we found that background noise addition was the one giving higher performance boosts. We discarded other techniques like time stretching or pitch shifting, as their gains were not stastically significant in very noisy scenarios. We combined different noise recordings (conversations, office and living room ambiences, TV and music) with clean speech in a wide range of SNRs ($[5, 30]$ or $[-10, 50]$ dB SNR). Every epoch we randomly selected a background noise sample per speech utterance, and combined them with the SNR from a randomly chosen SNR range.

Wake-Up Word Detection Models

To evaluate the effectiveness of task-aware speech enhancement models, we examined how they affect trigger word detection in various models. The devices that typically run these models have limited capabilities, so it is important to consider the time it takes for audio to be processed. While larger SE models generally perform better, they may also cause unwanted delays in detection, which can degrade the overall user experience throughout the conversation.

To establish a starting point for classification, we utilized the well-known convolutional neural network (CNN) called `LeNet` (LeCun et al., 2015). It is composed of two convolution layers with ReLU activations and two pooling layers, followed by two fully-connected layers.

Moreover, we took inspiration from the work of Sainath and Parada (Sainath and Parada, 2015), which explored lightweight CNNs for keyword detection by limiting the number of operations and parameters. We also employed Tang and Lin's re-implementation of this work in PyTorch (Tang and Lin, 2017), therefore using the `cnn-trad-pool2` architecture, which has two convolutional layers, each followed by time and frequency pooling. Additionally, Tang and Lin worked with deep residual networks combined with dilated convolutions (Tang and Lin, 2018), which yielded similar results to other CNN-based architectures while allowing for variations in depth and width to create small-footprint models. We employed three models from this work: `resnet15`, `resnet15-narrow`, and `resnet8`, which have 15, 15, and 8 ResNet blocks and 45, 19, and 45 feature maps, respectively.

Furthermore, we incorporated two RNN-based models, `SGRU` and `SGRU2`, which are based on the open source tool Mycroft Precise (Scholefield, 2019), a TensorFlow-based, lightweight WUW detection tool. We implemented these models in PyTorch, making larger variations of the original tool. `SGRU` had a single GRU with a hidden size of 200, while `SGRU2` had two GRUs with a hidden size of 100.

Lastly, we adapted an architecture proposed in Kaggle's FAT 2019 competition ("mhiro2", 2019) and called it `CNN-FAT2019`. This architecture has eight convolutional layers with ReLU activations and pooling layers every two convolutional layers. It demonstrated strong performance in tasks like audio tagging and detecting gender, identity, and speech events from pulse signals (Cámbara et al., 2020). This was the largest architecture we used in our study.

Table 4.8 displays the parameters, operations (multiplications and additions), and size of each keyword detection architecture used. RNN-based networks are the most compact, while ResNet-based architectures demonstrate varying amounts of operations and parameters based on their depth and width.

Classifier	**Parameters**	**Operations (mult. and add.)**	**Size (MB)**
lenet	4.7M	21M	19.2
cnn-trad-pool2	183k	42M	2.23
resnet15	237.4k	1433M	29.96
resnet15-narrow	42.4k	256M	12.44
resnet8	109k	57M	3.55
sgru	145.6k	144.4k	0.81
sgru2	103.4k	102.2k	0.53
cnn-fat2019	5.2M	1218M	41.9

Table 4.8: Parameters and number of operations of WUW detection models.

Training

We segmented speech utterances using a fixed window length of 1.5 seconds, which is generally sufficient to capture the average duration of the WUW (1.0 second) based on the SAD timestamps. We randomly mixed speech with background noise following the process outlined in Section 4.2.2 and using a given SNR range. The SE model was trained to handle a broad SNR range of $[-10, 50]$ dBs, while the WUW models were trained for two different scenarios: a classifier trained on the same SNR range as the SE model and a classifier with less noise awareness trained on a narrower SNR range of $[5, 30]$ dBs. This allowed us to

investigate the effect of the SE model on classifiers trained with different levels of noise.

To mitigate data imbalance, we used a weighted sampler to balance the classes in each batch. Additionally, we ensured that negative samples from the OK Aura dataset were always present in each batch through batching. This approach increased the presence of negative samples that were phonetically similar to the WUW during training, thereby enhancing their representation.

We used the loss function in Equation (4.1) to train various models in different ways. We defined different SE models and classifiers based on the loss function used. These models included:

(a) The Isolated classifier, where the autoencoder was removed from the architecture (Figure 4.3), and any of the classifiers were trained using the noisy audio as input with $\alpha = \beta = 0$ and $\gamma = 1$.

(b) The Separate SE and classifier model, where the classifier was removed from the architecture, and the autoencoder was optimized based on the reconstruction losses only with $\alpha = \beta = 1$ and $\gamma = 0$.

(c) The Task-Aware SE (TASE) model, where only the operations of a frozen pretrained classifier were backpropagated to the SE model, which was optimized with the reconstruction losses altogether with $\alpha = \beta = \gamma = 1$.

(d) The End-to-End TASE (TASE-E2E) model, where the autoencoder and classifier were trained jointly using the three losses with $\alpha = \beta = \gamma = 1$.

To train the models, early stopping was used based on the validation loss, with 60 epochs of patience, for a maximum of 200 epochs. If there was no improvement in 20 consecutive epochs, the learning rate was decreased by an order of magnitude. The Adam optimizer was used with a starting learning rate of 10^{-4} for the E2E case and 10^{-3} for the others, with a batch size of 50.

Testing

The test results were based on a binary classification task, where it was determined whether if the WUW was contained in a single time window or not. To create the test data, both negative and positive samples were mixed with background noise at a specific SNR level. Youden's J statistic (Youden, 1950) was used to determine the decision threshold based on the output probabilities from a model. F1-score was then computed for analysis and comparison across all WUW

classifiers and SNR ranges. Objective metrics PESQ (Rix et al., 2001) and STOI (Taal et al., 2010) were also reported on the Valentini et al. benchmark dataset (Valentini Botinhao et al., 2016) for comparison. The dataset contains clean and noisy speech in English with 15 different background noises. Two seconds of each enhanced audio clip were randomly selected from the noisy test set for PESQ and STOI measurements. Further details can be found in Section 4.2.2.

4.2.3. Results

The TASE architecture was used to enhance an audio sample that includes background music and a keyword spoken between 1.15 and 1.95 seconds. Figure 4.4 displays the spectrogram of the enhanced audio, which contains all the relevant speech information necessary for subsequent classification.

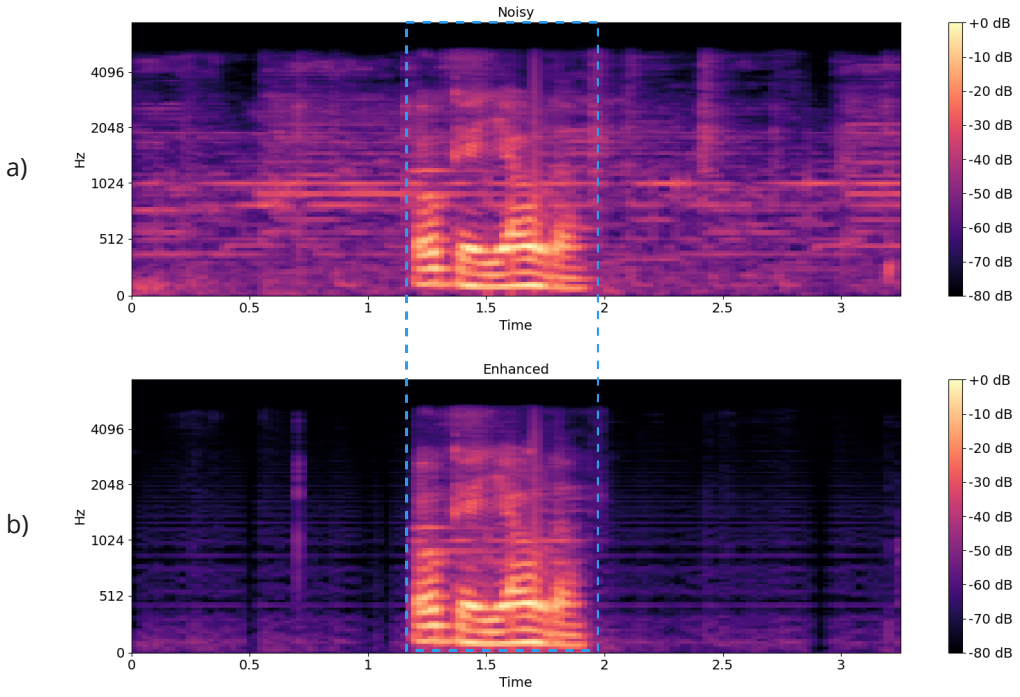
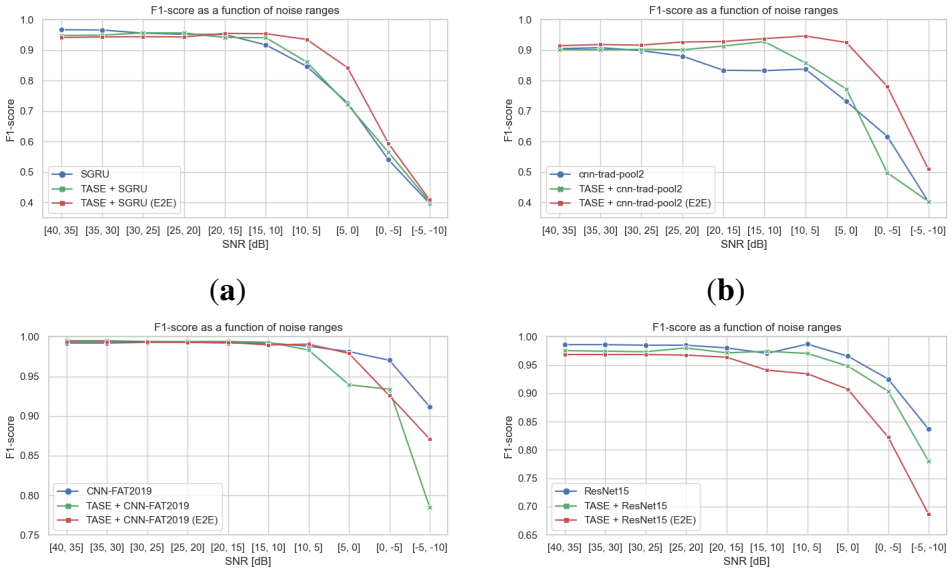


Figure 4.4: Example of Speech Enhancement spectrograms. Each figure shows (**a**) a noisy log-mel spectrogram and (**b**) an enhanced log-mel spectrogram. The blue rectangle shows where the "OK Aura" keyword is placed.



(**a**) (**b**)

(**c**) (**d**)

Figure 4.5: WUW detection performance comparison for different models in terms of F1-score, with and without TASE. All models are trained in the range of $[-10, 50]$ dB SNR. TASE is not beneficial in noisy scenarios for large architectures (bottom row), while it does contribute positively to smaller models, especially when trained jointly end-to-end (upper row). (**a**) SGRU. (**b**) cnn-trad-pool2. (**c**) CNN-FAT2019. (**d**) ResNet15.

Figures 4.5a–d display the performance of the TASE architecture when paired with various WUW classifiers described in Section 4.2.2. Our results indicate that TASE greatly benefits models such as SGRU and cnn-trad-pool2, which exhibit low resistance to noise compared to ResNet15 or CNN-FAT2019. However, TASE provides equal or worse results for ResNet15 or CNN-FAT2019 at certain noise levels. We believe that ResNet15 and CNN-FAT2019, being larger and more complex architectures, may not benefit as much from speech enhancement because they already handle noise nuances more accurately. We acknowledge that we did not fine-tune the hyperparameters extensively for every architecture due to computational constraints, and thus, our default hyperparameter selection may be biased toward specific architectures, resulting in lower performance for TASE-E2E in ResNet15. Further information about the metrics for SGRU and cnn-trad-pool2 can be found in (Cámbara et al., 2022).

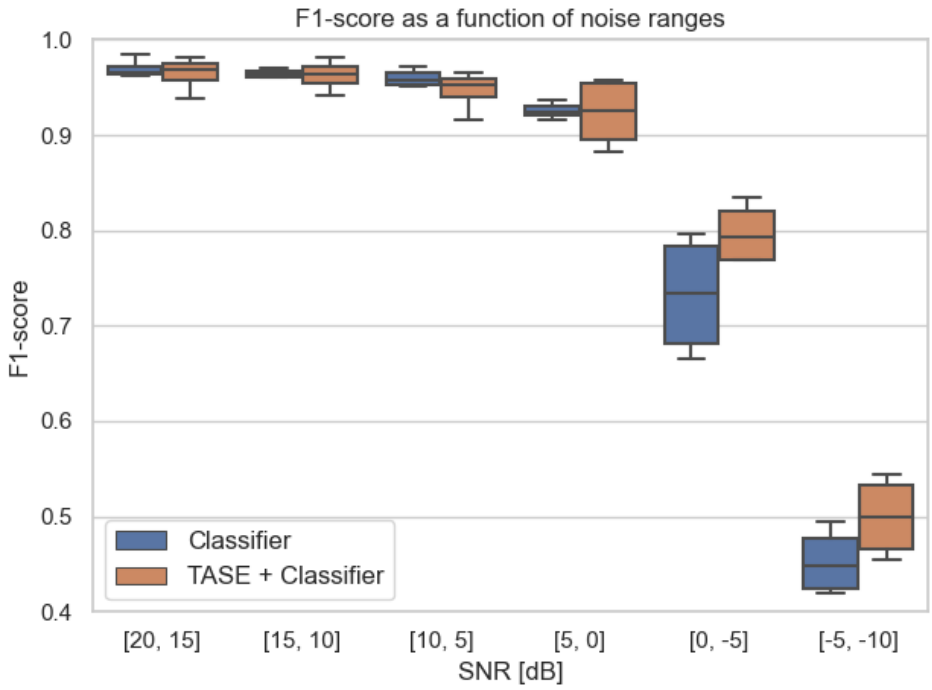


Figure 4.6: F1-score box plot for different SNR ranges. Classifiers trained with a limited range of noise ($[5, 30]$ dB SNR).

In addition, Figure 4.6 illustrates the enhancement in detecting WUW in noisy environments by combining our TASE model with other classifiers described in Section 4.2.2, which are neither large nor robust to noise (SGRU2, ResNet8, ResNet15-narrow), and LeNet, which architecture has not been optimized for audio tasks. The classifiers were trained with low noise ($[5, 30]$ dB SNR) to simulate a voice assistant system that has not encountered excessive amounts of noise during training. When SE is applied in quiet scenarios, the results remain relatively good, and particularly, the models are improved in lower SNR ranges.

However, if the classifiers are trained with a wider range of SNR ($[-10, 50]$ dB SNR) using data augmentation, the difference in performance when using TASE is notably reduced. F1-scores for both options are similar for most SNR ranges. Although there is a slight advantage for the model over TASE in the noisiest range of $[-5, -10]$ dB SNR, it is not as large as the improvement reported in Figure 4.6. See Figure 4.7.

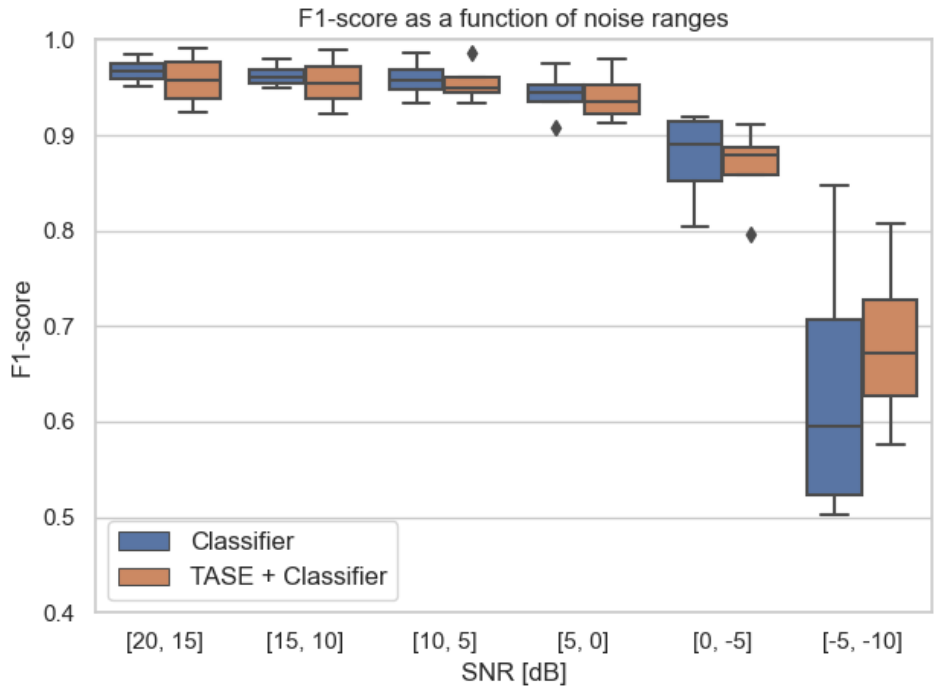


Figure 4.7: F1-score box plot for different SNR ranges. Classifiers trained with a very wide range of noise ($[-10, 50]$ dB SNR).

The loss function parameters (4.1) for training the classifier and SE model have been defined in Section 4.2.2, with three different training approaches: standalone, coupled with the classifier, and end-to-end training. Figure 4.8 compares the performance of these cases using a LeNet as a WUW detector. The TASE-E2E case outperforms all other cases in almost every SNR range, while the results are similar for the four models from 40 dB to 10 dB of SNR. The classifiers without SE model perform worst in the noisiest ranges, followed by the separate SE case where only the waveform and spectral reconstruction losses are used. The TASE case, which includes the classification loss in the training stage, improves the WUW detection results, but the best results are obtained with the TASE-E2E case, where the SE models and the classifiers are jointly trained using all three losses.

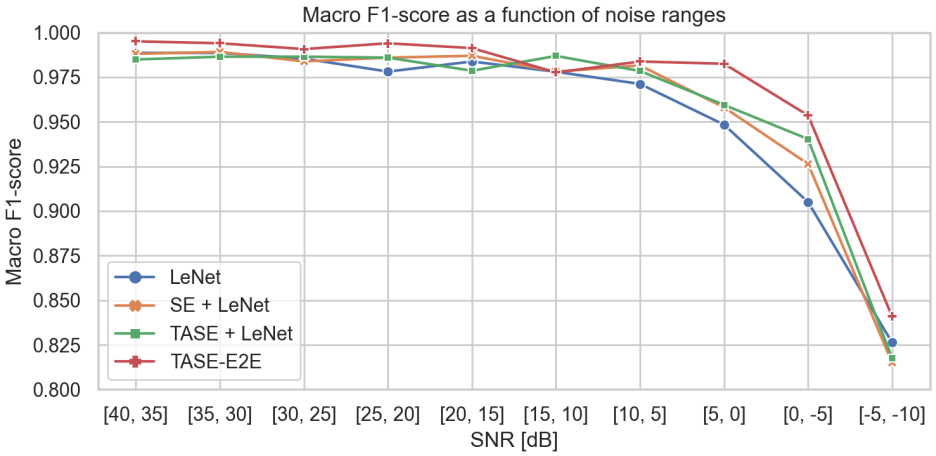


Figure 4.8: Comparison of different training methods for the SE models and LeNet classifier, in terms of the macro F1-Score for different SNR ranges. All models trained in the range of $[-10, 50]$ dB SNR.

The WUW detection performance of TASE-E2E is compared to other state-of-the-art SE models (SEGAN (Pascual et al., 2017) and Denoiser (Défossez et al., 2020)). We employed a data augmented LeNet in different noise scenarios. Table 4.9 shows that our setup outperforms other more powerful but more general SE models when training the models together with the task loss. We hypothesize that this is because the SE model naturally adapts to the classifier during end-to-end training and has been trained with a focus on common home noises. TASE-E2E improves detection against using no SE, particularly in scenarios with background conversations, loud office or TV noises, as shown in Table 4.10.

The results presented in Table 4.11 reveal that our SE system does not enhance speech quality when compared to the scenario where no model is used to improve speech. This outcome was expected, given that our models were not trained to remove generic background noises present in the Valentini dataset. Instead, the SE system was designed to eliminate background conversations and TV noise that could trigger the device, thereby leading to speech degradation. However, we did notice that the PESQ metric improved in the case of TASE combined with a LeNet classifier in comparison to SE. Furthermore, the best results were obtained with the end-to-end approach, where the PESQ and STOI scores were maintained at similar levels to those obtained without an SE module. These results demonstrate that incorporating the classification task in the loss function enhances the ability of the SE model to clean speech.

SNR [dB]		No SE	SEGAN	Denoiser	JointSE
$[20, 10]$	Clean	0.980	0.964	0.980	**0.990**
$[10, 0]$	Noisy	0.969	0.940	0.955	**0.972**
$[0, -10]$	Very noisy	0.869	0.798	0.851	**0.902**

Table 4.9: Macro F1-score enhancing the noisy audios with state-of-the-art SE models and using a LeNet as a classifier.

SNR [dB]		Music	TV	Office	Living Room	Convers.
$[20, 10]$	Clean	1.0	-0.9	1.4	0.4	**2.3**
$[10, 0]$	Noisy	0.0	-1.2	0.8	0.4	**1.9**
$[0, -10]$	Very noisy	0.5	3.9	**11.2**	3.1	3.8

Table 4.10: Macro F1-score percentage difference between JointSE and LeNet without SE, for different background noises.

Architecture	PESQ	STOI
None	2.02	0.93
SE	1.89	0.93
TASE	1.97	0.92
TASE-E2E	2.02	0.93

Table 4.11: Objective evaluation of speech quality.

4.2.4. Conclusions

We believe that our study was the first to investigate the use of neural-based speech enhancement for wake-up word detection, and we showed that it improves classification performance. Additionally, we introduced a method for making the speech enhancement module aware of the WUW task by incorporating the wake-up word classification loss during training, which we call task-aware speech enhancement (TASE). TASE provides even better results than training the speech enhancement and wake-up word classification modules separately, and it can be accomplished by freezing the wake-up word module during speech enhancement training or by training both together from scratch, which we refer to as end-to-end task-aware speech enhancement (TASE-E2E). TASE-E2E achieved the best

classification performance among all the tested setups. Our experiments showed that the benefits of speech enhancement are particularly noticeable at noisier SNR ranges, between 10 and -10 dBs. We also compared the effectiveness of TASE to a standalone wake-up word classifier trained on a wide range of SNRs between 50 and -10 dBs. The results showed that TASE was equally effective or slightly better than not using it in severely noisy setups between -5 and -10 dBs SNR. Thus, TASE has the potential to improve the performance of standard neural net classifiers that are not specifically trained to be resilient to noise, and we encourage further research into the comparison between speech enhancement and noise data augmentation techniques. Finally, we suggest that future works investigate the particular challenges and issues that may arise in online streaming scenarios, given that we worked with segmented audio.

Chapter 5

LEVERAGING PHONETIC INFORMATION FROM A SELF-SUPERVISED MODEL

Neural-based ASR models keep improving their transcription quality, as bigger models that train with bigger data are created. However, the data acquisition process for speech recognition is very costly, not only because of obtaining the speech audio, but mostly due to the effort in transcribing it. Thus, many recent findings in ASR have been related to the application of self-supervised learning (SSL) algorithms to it. SSL consists in designing training pipelines where ground truth labels are automatically generated. Masked language models are an example of this paradigm, as they learn by predicting masked parts of a text sentence, something that it is done by randomly sampling words to be masked, without explicit human action. Of course, we could think of similar tasks done with speech data, something that has been studied in works like CPC (Oord et al., 2018), PASE (Pascual et al., 2019) or wav2vec2.0 (Baevski et al., 2020).

From the aforementioned models, wav2vec2.0, altogether with HuBERT, was the highest performing one in several speech tasks, as recorded in SUPERB benchmark (Yang et al., 2021), for a period of time. It also gained fast adoption, as its open-source code was given to the community in repositories like fairseq (Ott et al., 2019), HuggingFace (Wolf et al., 2019) and SpeechBrain (Ravanelli et al., 2021). This model, along with the whole SSL research field, was of interest for our work, as the core idea of speech SSL is in line with our aim, which is enhancing features for a better recognition. Wav2vec2.0 delivered impressive results for ASR with only few transcribed data for fine-tuning, but its feature encoder was big in size and not straightforward to use in small devices, where also small la-

tency is a must. We knew that wav2vec2.0, during training, was learning a vector-quantized codebook of pseudophonemes, that it used as targets in its SSL scheme. This codebook was only used at training, to make the encoder learn about such phonetic information, and then thrown away for inference. We hypothesized that this phonetic information contained in the codebook could be also useful for training speech recognition classifiers, and its smaller size (it was only a matrix, in the end) could be more suitable for low-latency devices (such as Aura, for instance). For such a reason, we decided to explore its usage on a constrained speech recognition setup: keyword spotting. Particularly, we found a very interesting architectural synergy between wav2vec2.0 latent codebook and a cross-attention based model proposal: the Perceiver architecture (Jaegle et al., 2021).

Being so, we discovered an efficient method to use the linguistic knowledge from a pretrained wav2vec2.0 model for small footprint Keyword Spotting (KWS). Instead of using the encoder with over 95 million parameters, we repurposed the phonetic information in the latent codebook, typically discarded after pretraining. By transferring the codebook as weights for the latent bottleneck of a Keyword Spotting Perceiver (Jaegle et al., 2021), the model is initialized with phonetic embeddings. The Perceiver design employs cross-attention between these embeddings and input data to generate improved representations. This approach offers accuracy improvements compared to random initialization without increasing latency. Furthermore, we demonstrated that the phonetic embeddings can be downsampled using k-means clustering, accelerating inference by 3.5 times with only minor accuracy penalties.

5.1. Recycle Your Wav2Vec2 Codebook: a Speech Perceiver for Keyword Spotting

In this section, we start by diving deeper into the motivation behind our research. Then, we outline our proposed model, detailing the integration of wav2vec2.0 with the Perceiver model. Additionally, we discuss various techniques employed to enhance latency.

5.1.1. Motivation

Recent improvements in keyword spotting (KWS) consisted in using the Transformer architecture (Vaswani et al., 2017) and self-supervised learning proposals like wav2vec2.0 (Baevski et al., 2020). Transformers have an advantage over

CNNs and RNNs because they can capture information from wider contexts beyond a local range, and they avoid the issue of vanishing or exploding gradients. However, this advantage comes at a high computational cost due to the self-attention mechanism (Bahdanau et al., 2014), which is especially pronounced in high-dimensional modalities like audio or image. Keyword Spotting Transformer (KWT) (Berg et al., 2021) and Audio Spectrogram Transformer (AST) (Gong et al., 2021) models address this issue by using a method inspired by the Vision Transformer (ViT) (Dosovitskiy et al., 2020): downsampling the spectrogram into patches.

Meanwhile, models such as Wav2KWS (Seo et al., 2021) or the classifier from SUPERB (Yang et al., 2021) have effectively utilized wav2vec2.0 for Keyword Spotting (KWS). At training, wav2vec2.0 builds a latent codebook that captures phonetic information. These codes are used as a target to train its feature encoder. After training, the codebook is usually discarded, and only the encoder is used for downstream tasks such as KWS or ASR. Although the encoder can extract detailed features from raw waveforms, its large size (no less than 95 million parameters for the BASE model) and additional latency make its application to small KWS classifiers not so straightforward.

Thus, we concentrate on investigating ways to reuse the phonetic information stored in the wav2vec2.0 latent codebook. Our research demonstrates that this information can improve the accuracy of a KWS model at the start of the training process and results in improved convergence, with no added cost in latency and model parameters. Our proposal is based on a natural combination of the wav2vec2.0 and Perceiver models (Jaegle et al., 2021). The Perceiver model employs cross-attention between the input data and a smaller latent bottleneck tensor, which results in lower computational costs than pure self-attention over input data. We discovered that a pretrained wav2vec2.0 latent codebook can be used as an initialization for the Perceiver's latent bottleneck tensor, leading to improved accuracy compared to random weight initialization. Additionally, since many vectors in the wav2vec2.0 codebook contain similar phonetic information, we employ k-means clustering and average vectors from the same clusters, resulting in downsampled latent bottlenecks that provide faster inference with only a minor reduction in accuracy. This work's main contribution is twofold. Firstly, it sheds light on the effectiveness of utilizing latent codebook recycling for efficient transfer learning of the wav2vec2.0 model. Secondly, it demonstrates the application of the Perceiver model, for the first time to our knowledge, in a specific speech task such as Keyword Spotting (KWS).

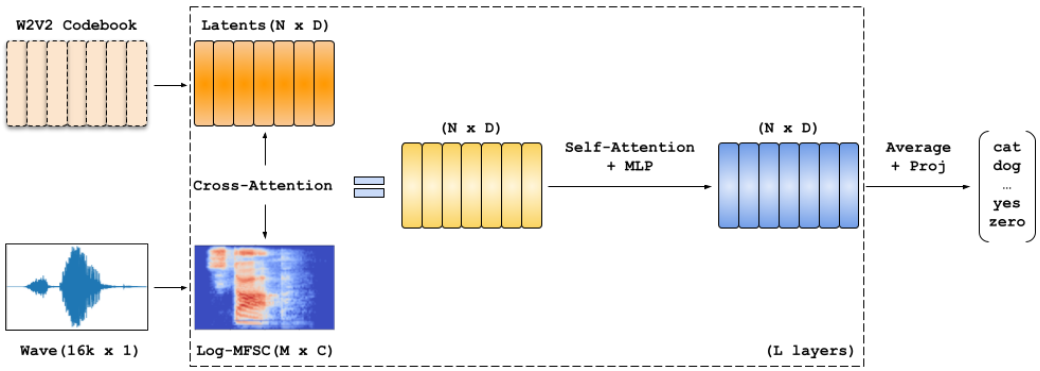


Figure 5.1: The Keyword Spotting Perceiver (KWP) model.

5.1.2. Model Description

The Keyword Spotting Perceiver (KWP) we created is configured to process inputs of 1-second waveforms, which are then transformed into log-mel spectrograms with 100 time steps (M) and 64 frequency bins (F). These spectrograms are then linearly projected into a 192-dimension space (C), creating an MxC data array. We append Fourier positional encodings to this data array along the C dimension, using 64 frequency bands and an optimal resolution of 224, as advised by the original Perceiver study. The model then engages in cross-attention between the data array and a latent bottleneck array of dimensions NxD, using a single attention head. The resulting output is further processed using a Transformer block that includes self-attention with eight heads and a multilayer perceptron (MLP) with a hidden size of 1024. The dimensions for both self-attention and cross-attention heads are fixed at 64. As the final output is another NxD latent array, we iterate the cross-attention process with the data array and the Transformer blocks for a total of 6 layers. We share weights across these layers following the pattern of a RNN. We initially experimented without sharing weights, but found it led to a decrease in performance due to overfitting. In the end, we average the latents across the D dimension, normalize the layer, and perform a linear projection to obtain the class logits for prediction. A visual model is provided in Figure 5.1.

The latent array could be randomly generated (KWP-BASE), or alternatively, we could use the weights from a pretrained wav2vec2.0 model's latent codebook (KWP-W2V2). For this research, we repurposed the latent weights from the HuggingFace repository's wav2vec2.0 BASE model[1]. The codebook was made up of 640 vectors (N) each with a dimension of 128 (D).

However, the computational challenge of cross-attention between the latent

[1]https://huggingface.co/facebook/wav2vec2-base

and data arrays is proportional to $O(MN)$, which negates the efficiency benefits when compared to self-attention over the data array $O(N^2)$. This is because $O(MN)$ equates to $O(100x640)$ or $(6.4x10^4)$, which is greater than $O(N^2)$ or $O(100^2)$ or (10^4). To counteract this, we investigated three techniques to downsize this latent space to smaller dimensions $N = [320, 160, 80, 40, 20]$ through: (1) selecting vectors at random, (2) pooling adjacent vectors for an average, and (3) using k-means clustering to average vectors within the same group. The wav2vec2.0 paper posits that the majority of the codebook latents are specific to English phonemes, with some phonemes being represented by multiple latents. For example, the silence phoneme accounts for 22% of the codebook. As a result, we expected that k-means clustering would be the most effective downsizing method by grouping similar or identical phoneme latents. Simple average pooling may retain phonetic data, but we anticipated it to be less optimal as we couldn't ensure that contiguous vectors in the codebook aligned with similar phonetics, which could lead to phoneme information blending. In contrast, random sampling ensures the preservation of each vector's individual information in the codebook, but as N decreases, a significant amount of information could be lost due to the exclusion of most vectors.

5.2. Accuracy and Latency of the Keyword Spotting Perceiver

In this section, we focus on evaluating the accuracy and latency of our Keyword Spotting Perceiver through a series of experiments. We will outline the experimental methodology used, examine the model's performance both at initialization and upon convergence, and conclude with key insights drawn from these assessments.

5.2.1. Methodology

Let's elaborate on the assessment carried out for our Keyword Spotting Perceiver proposal. Initially, we examined the impact of transferring the wav2vec2.0 latent codebook to the Perceiver bottleneck during the initialization phase. We also explored varying methods for downsampling this latent space and compared the accuracy of the baseline KWP-BASE model and the wav2vec2.0-initialized KWP-W2V2 model. Subsequently, we retained the most effective downsampling method for the next round of experiments, during which we allowed the system to train until it reached convergence. For KWP-BASE and KWP-W2V2 models with

different latent number variants $N = [320, 160, 80, 40, 20]$, we reported accuracy, model size, and inference time metrics.

We carried out the training, validation, and testing stages using the standard partitions from the Google Speech Commands V2 dataset (Warden, 2018), and the accuracy metrics were derived from the 35-commands task. Timing metrics were obtained by performing inference on 1-second waveforms using a CPU, with a warm-up period of 10 forward passes and an average time calculated over 150 forward passes. We employed the AdamW optimizer (Loshchilov and Hutter, 2018) with a step learning rate scheduler, reducing the learning rate each epoch by a gamma factor of 0.98, starting with an initial learning rate of $1e^{-4}$. The batch size was set to 32, with training occurring for a single epoch during the initialization experiments and 400 epochs during the convergence experiments. For the latter, we selected the top-10 checkpoints with the highest validation accuracy, averaged their weights to create the final checkpoint, which was then used for test accuracy measurements. The PyTorch code used for our experiments is available to the public[2].

In terms of data augmentation, we implemented time shifting of ± 0.1 seconds with a 60% probability. We also resampled the waveform signal within a $[0.85, 1.15]$ fraction of the input sampling rate, which was set to 16 kHz, with a 100% probability. We added background noise within a range of $[5.0, 30.0]$ dBs and applied SpecAugment (Park et al., 2019) with two time masks of 25 frame size and two frequency masks of 7 frames each. Both of these data augmentation methods were applied with a 100% probability during training. However, for the shorter initialization experiments that lasted a single epoch, we reduced the augmentation conditions to allow the system to learn more during the initial stages. The probabilities for time shifting and resampling were decreased to 30%, SpecAugment to 70%, and background noise addition to 80%.

5.2.2. Initialization with Wav2Vec2.0 Codebook

We tested the effect of transferring the wav2vec2.0 codebook to KWP during the initialization phase. We achieved this by calculating the test accuracy following a single epoch, and we carried out this process ten times, each with a different seed. We drew comparisons between KWP-BASE and KWP-W2V2, both containing all the $N = 640$ latent vectors. In this case, we allowed the latent bottleneck weights to be modifiable (BASE and W2V2) and also kept them constant (BASE-Frozen and W2V2-Frozen).

[2]https://github.com/gcambara/speech-commands

As illustrated by Figure 5.2 (left), both W2V2 and W2V2-Frozen displayed a notable performance edge over BASE and BASE-Frozen. This led us to infer that the phonetic information imported from the wav2vec2.0 latent codebook gave the model an initial advantage, providing useful information that the cross-attention mechanism could utilize from the get-go. It was also intriguing to note the lack of performance disparity between BASE and BASE-Frozen. Our theory is that, during the first epoch, the randomly initialized BASE model did not accumulate enough phonetic information in its latent bottleneck. This limited the ability of the cross-attention to exploit associations with input data. On the contrary, the W2V2 model was able to harness the power of cross-attention early, leading to a beneficial cycle between the phonetic data in the latent codebook and the cross-attention weights linked to input data.

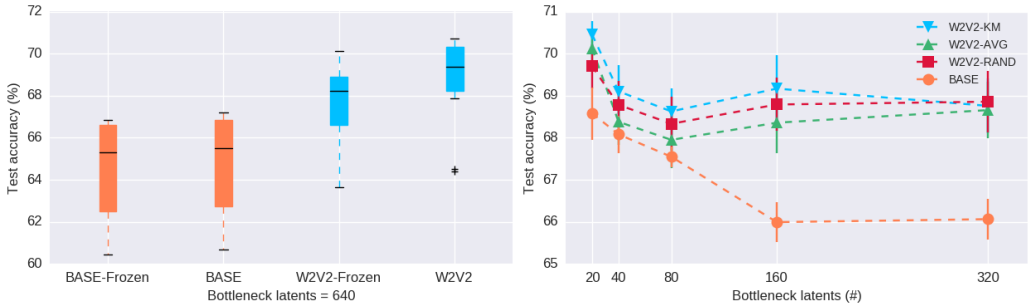


Figure 5.2: The test accuracy following a unique training epoch for two models - one randomly initialized (BASE) and the other initialized with wav2vec2.0 latent codebook weights (W2V2), with either adaptable or static weights (left). Also, we present the outcomes of bottleneck latents downsampling using k-means clustering (KM), average pooling (AVG), and random sampling (RAND) (right).

We proceeded by replicating the same ten-seed experiment for $N = [320, 160, 80, 40, 20]$ latent vectors in the bottleneck. We tested different downsampling techniques: k-means clustering (W2V2-KM), average pooling (W2V2-AVG), and random sampling (W2V2-RAND). The outcomes, depicted in Figure 5.2 (right), underscored the efficacy of all three latent downsampling methods in improving performance compared to the BASE model. Of these, W2V2-KM emerged as the top performer. This validated our belief that averaging latents associated with the same phonetic clusters was a superior strategy compared to the simple averaging of contiguous latents, as done in W2V2-AVG, or the random sampling of latent vectors, as performed by W2V2-RAND. The latter method, especially, saw a decrease in representational capacity as N decreased.

5.2.3. Assessment at Convergence

In order to assess the precision of the KWP-BASE and KWP-W2V2 models post-convergence, we let the models undergo a prolonged training session of 400 epochs. At this stage, we only experimented with adaptable latent weights and k-means clustering downsampling since the latter proved to deliver superior initialization results. We carried out training and testing with 3 seeds, changing the number of latents once again with the same range, $N = [640, 320, 160, 80, 40, 20]$, and drew comparisons between BASE and W2V2 versions.

As shown in Figure 5.3, the W2V2 version maintained a considerable lead across all the latent numbers, reaching a peak mean accuracy of $96.26 \pm 0.04\%$ with 640 latents. This exceeded the BASE's highest accuracy of $95.6 \pm 0.2\%$ achieved with 80 latents. It appears that the W2V2 variant scales well with the number of latents, unlike the BASE model, which might find it challenging to cluster phonetic information in the latent space as it expands. However, KWP (1.5M parameters) still falls a bit short compared to its self-attention counterparts, with the lightest KWT (0.6M parameters) achieving a 96.8% accuracy, and AST reaching 98.1%. It is worth mentioning that AST is pre-trained with ImageNet (Deng et al., 2009) and has a significantly larger size (87M parameters). Despite this, we encourage further exploration into fine-tuning KWP to achieve state-of-the-art performance.

The inference time for the model with 640 latents stands at 49 ± 5 ms, and for the smaller model with 20 latents, it's 14 ± 2 ms. Considering that the accuracy is $95.3 \pm 0.1\%$ for the latter, we see a relative accuracy loss of only 1% with k-means clustering downsampling while boosting the inference speed by 3.5 times. The accuracy of the BASE model at 20 latents is $94.6 \pm 0.3\%$, which is significantly lower than that of W2V2. This proves that even a severe downsampling from 640 to 20 latents of wav2vec2.0 information remains a more effective choice compared to randomly initializing the latent space in KWP.

5.2.4. Conclusions

In the study presented herein, we unveiled the potential of reusing the phonetic information embedded in the wav2vec2.0 latent codebook by transplanting it into the latent bottleneck weights of a Keyword Spotting Perceiver. By doing so, we noted a substantial and consistent increase in accuracy compared to scenarios where the latent bottleneck was randomly initialized. This improvement was not just observed during the initial stages of training but was also sustained during the later phases of the Keyword Spotting task.

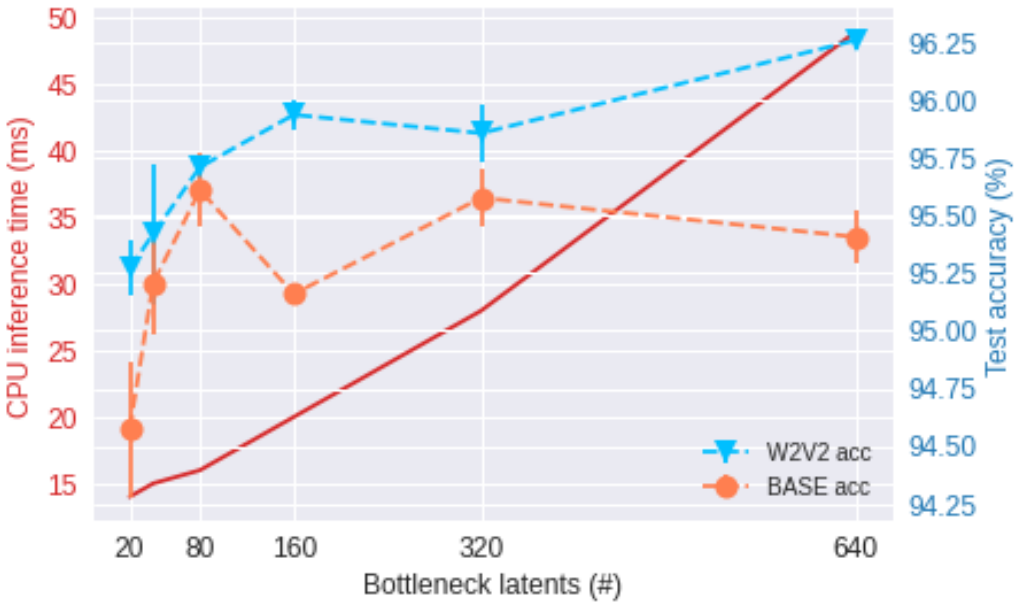


Figure 5.3: Post-convergence test accuracies for KWP-BASE (depicted in orange) and KWP-W2V2 (shown in blue) with varying counts of bottleneck latents, alongside the CPU inference time (represented in red).

Furthermore, our exploration extended into devising efficient downsampling strategies to compress the latent codebook. We found that simply averaging k-means clusters led to notable enhancements in model performance. In fact, this technique enabled us to reduce the inference time of the model by up to 3.5 times, which resulted in only a marginal 1% decrease in accuracy.

The implications of our findings are substantial, as they underscore the value of effectively leveraging the rich information contained within large self-supervised models like wav2vec2.0. We firmly believe that our work serves as a robust foundation and a catalyst for future research. Such future endeavors could explore more efficient ways to utilize the wealth of information in these large models or even extend the application of our techniques to other tasks such as large vocabulary ASR. Our work also promotes the exploration of different methods to transfer knowledge between models, beyond our particular exploration on wav2vec2.0.

Chapter 6

CONCLUSIONS AND FUTURE WORK

I started this thesis by stating the motivation on working towards the challenges of deep learning methods in the realm of automatic speech recognition, particularly at the feature representation level. While it is undeniable that deep learning has propelled the field forward, the performance of speech recognition systems is hindered by the lack of data, the requirement of small footprint devices or the presence of acoustically challenging environments, just to name a few sources of issues. Out of the research labs, many real-world systems and projects operate under some of these conditions.

The work in this thesis has been focused to specific contributions on three different research threads that branch from the field of feature enhancement for ASR. Firstly, I explored the integration of prosody and voice quality features into neural models for speech recognition. Given the inclination of deep learning towards diminishing additional features, we hypothesized that, particularly in low data situations with less than 1000 hours, incorporating prosodic context could be advantageous. The experimental work on this thread resulted in the development of new convolutional and Transformer models, which were shared through publications and open-source contributions.

Subsequently, we addressed challenges related to speech and noises in a project with Telefónica Research, focusing on a subset task of speech recognition: wake-up word detection. Herein, we investigated optimal strategies for applying speech features enhancement to improve the robustness of wake-up word recognition models, which yielded conference and journal publications and the release of an open-source wake-up word dataset.

Lastly, motivated by data limitations and the requirement for small compact

models, we explored self-supervised learning models for speech. Typically, most of the self-supervised learning approaches involve the usage of large encoder models, that are not always suitable for small footprint devices. In this occasion, we investigated on ways to leverage the knowledge absorbed by these models but in a zero-cost way regarding latency and model parameters. For such work, aimed for speech keyword spotting, we also published a conference paper and delivered open-source code.

The outline of this chapter is as follows: I present final reflections and conclusions on the experiments done in this thesis in Section 6.1, and in Section 6.2 I outline next steps for future research. Lastly, I summarize the accomplishments and attributions resulting from this work in Section 6.3.

6.1. Conclusions

Feature enhancement for speech recognition can be achieved in various ways. In recent years, the field that has seen the most prominent research has been self-supervised learning for speech. We conducted research in this area, focusing primarily on leveraging information encapsulated by self-supervised learning models in a manner that is cost-effective in terms of latency and computation for speech recognition modules. Nevertheless, we also explored other research streams that are not only independent but also complementary to self-supervised learning techniques. Specifically, we investigated various ways to couple speech enhancement modules with wake-up word classifiers to improve detection in noisy home environments. Additionally, we focused on exploring the use of prosody and voice quality measurements to complement spectral features, assessing their utility in setups with fewer than 1000 hours of data. Research across these three streams provided us with valuable insights into different ways to enhance features for improved performance in speech recognition tasks, whether they involve large vocabulary speech recognition, keyword spotting, or simply wake-up word detection.

The first research stream we explored to enhance features for speech recognition involved the application of pitch and voice quality features, specifically jitter and shimmer. We were motivated by the success of applying such features to other speech tasks, such as speaker recognition and diarization. Moreover, prosody features like pitch had already been successfully applied to speech recognition in classic models before the advent of deep learning. We hypothesized that such features could still be beneficial for deep learning systems, as they could provide more specific information that might assist during model training, especially in situations where large datasets are not available.

Thus, we did an initial probe into the implications of utilizing pitch, along with jitter/shimmer voice quality measurements, in the context of large vocabulary speech recognition, modeled by convolutional neural networks. Using a publicly accessible Spanish speech corpus, our experiments exhibited improvements in model robustness, effecting a relative 7% diminution in WER under particular circumstances. Furthermore, we implemented these feature extraction capabilities in the wav2letter code, allowing for the straightforward replication of our results or the direct implementation of pitch and voice quality features to wav2letter models. Additionally, we supplied the recipe for the Common Voice Spanish dataset, representing the first of its kind suitable for wav2letter using a publicly available Spanish dataset. Also, the recipe facilitating LibriSpeech experiments was published, with improvements up to a relative marking an upsurge to a 2.94% in WER.

It should be noted that the method used for feature combination was straightforward, simply concatenating such features to the spectral ones at the input layer, without significant post-processing or adaptation of the model architecture. Given this, it was plausible to believe that additional enhancements could be made in applying pitch and voice quality measurements to cutting-edge neural models. Potential strategies included modifying the feature concatenation, perhaps by allocating specific filters to the new pitch and voice quality features. This adjustment was implemented in subsequent work, where we also transitioned from a convolutional to a Transformer-based architecture, demonstrating improved efficacy.

In this new study, we once again researched the usage of pitch and voice quality features, this time applying them to a state-of-the-art Transformer-based acoustic model. With the knowledge gained from previous research, we tried two different methods to apply the new features: first, by simply concatenating them to the spectral ones, and second, by processing them with separate convolutional filters. It turned out that, as we had hypothesized, the latter approach outperformed the former, with higher WER improvements (up to 5.6%) whether we applied pitch and voice quality features separately or combined them altogether. By using separate convolutional filters, we could control the total amount of pitch and voice quality features compared to the spectral ones. Whereas before we would have a single stream of 40 spectral features plus as many as 5 pitch and voice quality ones, now, with the separate filters, we could control it to have a proportion of 64 pitch and voice quality features against 192 spectral ones. This would make prosody information more explicit to the attention mechanisms in the Transformer. Such results suggested the potential of voice quality measurements to complement pitch features, yielding better recognition even for a simple benchmarking task of speech recognition for English. We believe that these features could be even more useful for tonal languages like Chinese, or speech recognition involving punc-

tuation marks, for instance. Moreover, our explorations only delved into features derived from pitch, but it could be interesting to explore feature enhancement with other prosody-related features like rhythm or intensity.

After exploring the addition of prosody features in situations where large datasets of speech were not accessible, we conducted research on another thread closely related to feature enhancement: the denoising of speech features as a preliminary step before recognition, in order to better manage challenging acoustic environments. The first stepping stone in this research stream was the Albayzin 2020 challenge, where we had to develop a competitive speech recognition system for Spanish TV shows. Recognizing speech in TV shows was challenging because it was mixed with a plethora of acoustic events, such as overlapping speech, music, indoor and outdoor noises, etc. To tackle this, we applied speech enhancement models to denoise the highly complex audio present in TV shows.

Our primary discovery revealed that, while the models utilized for enhancement (Demucs and Denoiser) successfully purified audio from noise, they inadvertently introduced additional artifacts into samples that were initially clean, subsequently elevating the average WER slightly, which was not a desirable outcome. A closer examination of WER reductions across various SNR intervals indicated that the process of cleaning speech resulted in improved WERs when executed on noisy audio. Given our utilization of these ready-made denoisers and source separation models, we deduced that speech enhancement for speech recognition would be more effectively accomplished with a customized approach. This might involve activating speech enhancement models at lower SNR ranges, or alternatively, training the speech recognition models to disregard artifacts introduced by speech enhancement modules. We would delve deeper into the latter concept in our subsequent work, which explored applying speech enhancement for wake-up word detection in a real-world voice assistant, Aura.

During this new research project for Aura, we designed and collected a new database containing the wake-up word ”OK Aura”, plus some other sentences with similar, distracting phrases, such as ”OK Laura”. This dataset was named the ”OK Aura Dataset” and was publicly distributed under an End-User License Agreement. Having collected the new data, we initiated our novel study on the use of neural-based speech enhancement for wake-up word detection, considering the lessons learned in the Albayzin 2020 challenge. This time, we proposed a strategy, named task-aware speech enhancement (TASE), that informs the speech enhancement module about the wake-up word task by integrating the wake-up word classification loss during its training. TASE, offering superior results compared to independently training the speech enhancement and wake-up word classification modules, can be done by either freezing the wake-up word module during speech enhancement training or initiating training from scratch for both, the latter be-

ing dubbed end-to-end task-aware speech enhancement (TASE-E2E). TASE-E2E surfaced as the top performer in classification across all evaluated configurations. Our experimental findings highlighted the particularly prominent advantages of speech enhancement within noisier SNR spans, specifically between 10 and -10 dBs. We also compared the efficacy of TASE against a standalone wake-up word classifier, both trained across an expansive SNR spectrum from 50 to -10 dBs. The outcomes demonstrated that TASE matched or marginally surpassed the effectiveness of the standalone wake-up word classifier in notably noisy environments, between -5 and -10 dBs SNR. Consequently, TASE has the capability to amplify the efficiency of conventional neural network classifiers, which are not inherently designed to handle noise, and we advocate for additional research exploring the comparison between speech enhancement and noise data augmentation methods. In closing, we recommend subsequent studies to delve into the specific challenges and problems potentially surfacing in live streaming contexts, considering our work was conducted with segmented audio.

The years dedicated to crafting this thesis coincided with the rise of self-supervised learning applied to speech. We could not overlook these innovations and aimed to contribute to this research thread during our work. Self-supervised learning models were designed to mitigate data requirement issues, excelling at training with limited labeled data by pretraining with a self-supervised objective on unlabeled speech utterances. However, many paradigms, such as wav2vec2.0 and HuBERT to name a few, relied on large encoders with millions of parameters to absorb all this implicit speech information. While these models are suitable in low-resource data scenarios, they might not be the best option for use-cases with computational constraints, such as in small footprint devices. Keeping that in mind, and especially after having worked on home devices like Aura, we decided to explore feature enhancement with self-supervised learning techniques. This time, our focus would be to find a way to leverage information contained in these pretrained models, but without additional costs in latency or computational requirements. With that objective, we chose speech command detection as a simple task to further explore our research scope.

We accomplished our goal by identifying a synergy between two state-of-the-art models: wav2vec2.0, a self-supervised learning model, and Perceiver, a model that uses cross-attention between a compressed latent space and the input signal. As the wav2vec2.0 model learned a phonetic vector-quantized codebook during the training phase, we discovered that we could use such a codebook as the initialization of the latent bottleneck weights in the Perceiver. We not only designed the first Perceiver applied to speech, the Keyword Spotting Perceiver, but also demonstrated the effectiveness of our weight transfer method from the wav2vec2.0 codebook to the Perceiver weights. We observed accuracy improvements when using

the codebook as initialization compared to randomly initializing the weights, not only in the early stages of training, where it was notably pronounced, but also at the model's convergence.

Moreover, we researched further by investigating potent downsampling approaches to condense the latent codebook. Our discovery that straightforwardly averaging k-means clusters yielded significant enhancements in model performance was paramount. This method not only accelerated the inference time of the model by up to 3.5 times but also incurred only a slight 1% drop in accuracy. Our discoveries hold interesting implications, highlighting the importance of leveraging efficiently the abundant information embedded in extensive self-supervised models like wav2vec2.0. Forthcoming investigations could delve into more proficient methods to exploit the information in these large models or broaden the application of our strategies to additional tasks, such as large vocabulary speech recognition. Furthermore, our work points towards exploring alternative techniques for transferring knowledge between models, going beyond our specific exploration of wav2vec2.0.

Little else remains to be said now that we have reached the end of this work, other than revisiting the initial questions that we posed at the beginning and summarizing the findings we have extracted from our research:

I. **Could the incorporation of prosody and voice quality characteristics contribute to enhancing existing spectral feature-based systems?**

   Our explorations of convolutional and Transformer-based architectures for 1000 or fewer hours of data suggest that, yes, there is a reduction in word error rates when using such additional features. Furthermore, processing these features and spectral ones separately with different convolutional filters is an effective method to increase the prosody information ratio in the feature space, which yields better performance than simply concatenating spectral and prosody features.

II. **What is the relationship between wake-up word detection performance and varying signal-to-noise ratios, and how does the application of speech enhancement affect this relationship? Additionally, what are the most successful approaches for co-training speech enhancement and wake-up word detection models?**

   Wake-up word detection, like other speech recognition tasks, underperforms when the signal-to-noise ratio is reduced, so speech enhancement models serve as a viable solution to mitigate this issue. However, enhancement models can adversely affect the performance of classifiers at high signal-to-noise ratios of the input signal, as they may introduce slight distortions

into clean speech. Consequently, we discovered that jointly training speech enhancement and wake-up word detection models renders the compound model robust across a wide signal-to-noise ratio range, from very noisy to clean speech. Freezing the wake-up word detection module and backpropagating its loss to the speech enhancement is already beneficial, but optimal results are obtained when both are trained without freezing.

III. **Is there an innovative method for utilizing the phonetic information embedded within a model such as wav2vec2.0 to enhance keyword spotting performance, all while avoiding extra computational burden and latency?**

Indeed, we discovered a method to repurpose the phonetic information contained in the wav2vec2.0 codebook, which resulted in performance improvements for keyword spotting without incurring additional latency and complexity. We utilized the phonetic quantized vectors from the codebook to initialize the latent weights of a Keyword Spotting Perceiver, achieving greater accuracy than with random initialization. Moreover, we demonstrated that we could compress these phonetic weights by employing k-means clustering, therefore enabling even faster inference with only a minimal penalization in accuracy.

By the end of this thesis, we have managed to explore and contribute to three research streams in speech recognition feature enhancement, providing new insights to the community, as well as producing additional outcomes such as published recipes, source codes, papers, and datasets. Before detailing all these contributions, we dedicate the following section to discuss and inspire further work.

6.2. Future Work

The insights obtained throughout the course of this thesis inspire additional research initiatives on the topics studied herein. These final thoughts suggest potential next steps for the explored research streams, while also considering some outcomes published by the community during this time, beyond our contributions in the thesis.

Given the favorable outcomes from introducing pitch and voice quality features, we believe it is important to extend this exploration, especially in specific instances where substantial data is unavailable. Although the prevailing trend appears to continue toward enhancing systems to be as end-to-end as possible,

by supplementing them with additional training data and computational capacity, with Whisper (Radford et al., 2023) exemplifying this paradigm, we advocate for the expansion of our methods in low-resource scenarios. This is particularly relevant for tonal languages, where prosody carries significant semantic weight, and in applications like predicting punctuation marks. Moreover, we learned that the processing choices for prosody and voice quality features are important. Improved results were observed when utilizing dedicated convolutional filters instead of stacking them with spectral features. Consequently, we might consider alternative designs that further capitalize on prosody aspects to enhance transcriptions. Why not employ cross-attention between prosody and spectral or linguistic features to derive better alignments? Or introduce explicit rhythm or intensity features for a more comprehensive prosody representation? As we can see, there are some ideas that would be worth to explore.

Regarding speech enhancement for on-device wake-up word detection, we studied a setup with fixed size windows of speech, but in real-life scenarios detection works with audio streaming. For such a reason, we would encourage further research on optimizing the enhancement-detection pipeline for these streaming scenarios. Our denoising pipeline had a regression loss, but there are other generative approaches that are more advanced, like GANs, flows or the late diffusion models. It would be worth it to try these paradigms for denoising, altogether with the speech recognition loss, to make more advanced pipelines. Furthermore, we collected a variety of interesting examples in the ”OK Aura” dataset, with a classification of distractor examples that were labeled with the phonetic similarity they had to ”OK Aura”. We did not have the time and scope to research on this, but we think it would be very interesting to explore phonetic disambiguation with this dataset. Plus, we also had other labels like speaker's accent or approximated room dimension, so these are other interesting metadata that could motivate some research on accent detection or bias, or treatment of reverberations.

Lastly, we found it intriguing that the phonetic information from the vector-quantized codebook in wav2vec2.0 could be repurposed to enhance keyword spotting in a Perceiver model. This insight could easily be extended to apply a Perceiver, or any model based on cross-attention, to large vocabulary speech recognition. While we used the wav2vec2.0 codebook, there are alternative models that leverage discrete phonetic representations too, such as HuBERT, and several recently proposed models in text-to-speech, which focus on predicting vector-quantized speech tokens. Some examples of these models include Tortoise (Betker, 2023), AudioLM (Borsos et al., 2023), and VALL-E (Wang et al., 2023). Not only could we try using these vector-quantized representations for speech classification, but we could also explore compressing these codebooks with k-means to determine whether we can reduce the complexity or codec-based text-

to-speech modeling with minimal impact on synthesis quality.

On and on, the insights given by our findings plus the recent advancements made by fellow researchers encourage a wide variety of research streams that we think are worth to explore. We sincerely hope to inspire future work in these areas.

6.3. Achievements and Attributions

The process of this research has brought tangible outcomes in the form of papers, datasets, project deliveries and open-source code for the community. These are listed in this section, as well as attributions to the collaborators and sponsors that participated in the course of this thesis.

6.3.1. Publications

All the findings mentioned in this work have been published, mostly in conferences and journals, and just a few in pre-print pages like arXiv.org. Besides, some of the outcomes and research models developed here have been used in other research streams and projects. Some examples of these are the use of data augmentation for improving ASRs on a low-resource language like Quechua, or the application of our ASR technology to voice-to-voice translators for emergency teams, under the European project INGENIOUS [1]. These contributions have also been published in papers and are listed here:

- Cámbara, G., Grivolla, J., Farrús, M., Wanner, L. *Automatic Speech Translation for Multinational First Responder Teams.* Proceedings of the 20th ISCRAM Conference – Omaha, Nebraska, USA May 2023.
- Cámbara, G., Luque, J. Farrús, M. *Recycle Your Wav2Vec2 Codebook: A Speech Perceiver for Keyword Spotting.* Proceedings of the 29th International Conference on Computational Linguistics. Gyeongju, Republic of Korea. October 2022.
- Zevallos R., Bel N., Cámbara G., Farrús M., Luque J. *Data Augmentation for Low-Resource Quechua ASR Improvement.* Proceedings of the InterSpeech. Incheon, Republic of Korea, September 2022.
- Cámbara G., López F., Bonet B., Gómez P., Segura C., Farrús M., Luque J. *TASE: Task-Aware Speech Enhancement for Wake-Up Word Detection*

[1] `https://ingenious-first-responders.eu/`

in Voice Assistants. (Feature Paper). Applied Sciences. Special Issue on IberSPEECH 2020: Speech and Language Technologies for Iberian Languages), 12(4), pp. 1974, 2022.

- Peiró-Lilja A., Cámbara G., Farrús M., Luque J. *Naturalness and Intelligibility Monitoring for Text-to-Speech Evaluation.* Proceedings of the Speech Prosody Conference. Lisboa, Portugal, May 2022.
- Cámbara G., Farrús M., Luque J. *Voice Quality and Pitch Features in Transformer-Based Speech Recognition.* Proceedings of the Speech Prosody Conference. Lisboa, Portugal, May 2022.
- Cistola, G., Peiró-Lilja, A., Cámbara, G., van der Meulen, I., Farrús, M. *Influence of TTS systems performance on reaction times in people with aphasia.* Applied Sciences, 11(23), 11320. 2021.
- Codina-Filbà, J., Cámbara, G., Luque, J., Farrús, M. *Influence of ASR and Language Model on Alzheimer's Disease Detection.* arXiv preprint arXiv:2110.15704. 2021.
- Cámbara G., Peiró-Lilja A., Farrús M., Luque J. *English Accent Accuracy Analysis in a State-of-the-Art Automatic Speech Recognition System.* PaPE 2021 Workshop "From speech technology to big data phonetics and phonology - a win-win paradigm". Barcelona, Catalonia, June 2021.
- Kocour M., Cámbara G., Luque J., Bonet D., Farrús M., Karafiát M., Veselý K., Černocký J. *BCN2BRNO: ASR System Fusion for Albayzin 2020 Speech to Text Challenge.* Proceedings of the IberSpeech. Valladolid, Spain, March 2021.
- Bonet, D., Cámbara, G., López, F., Gómez, P., Segura, C., Luque, J. *Speech enhancement for wake-up-word detection in voice assistants.* arXiv preprint arXiv:2101.12732. 2021.
- Codina-Filbà J., Cámbara G., Peiró-Lilja A., Grivolla J., Carlini R., Farrús M. *The INGENIOUS Multilingual Operations App.* Proceeding of the InterSpeech. Brno, Czech Republic, August 2021.
- Cámbara, G., Luque, J., Farrús, M. *Convolutional speech recognition with pitch and voice quality features.* arXiv preprint arXiv:2009.01309. 2020.
- Cámbara, G., Luque, J., Farrús, M. *Detection of speech events and speaker characteristics through photo-plethysmographic signal neural processing.* IEEE International Conference on Acoustics, Speech and Signal Processing. Barcelona, Spain. May 2020.

6.3.2. Datasets

Our work in speech enhancement for wake-up word detection was done under Telefónica's project "Aura", a voice assistant for homes. It implied the recording and publication of *OK Aura*, a speech dataset that contains 1247 utterances (1.4 hours) from 80 speakers. Speakers pronounce the wake-up word itself "OK Aura", plus other sentences that might be similar, or not, to "OK Aura".

This dataset contains rich metadata annotations, so it is possible to study diverse factors and biases that might affect wake-up word detection performance: accent, gender, prosody/emotion, room size, distance to the microphone, etc. Besides, it also contains recordings of sentences that are phonetically similar to "OK Aura", like "Porque Laura..." or "... como Aura...", with the purpose to experiment with difficult sentences. The full dataset is available in Zenodo [2] through an End-User License Agreement.

6.3.3. Open-Source Code

Most of the software developed during this thesis has been published open-source, so it is reusable by the community for other projects or experiment replication. Some exceptions being the source code for some developments of Aura technology, which is property of Telefónica, or specific ASR components for the INGENIOUS project. Besides that, most of the code used here is public, and also I have done contributions on open-sourcing some of the state-of-the-art baselines used in the thesis, like the wav2vec2.0 implementation that I co-authored in the SpeechBrain toolkit.

- **speech-commands** [3] - An implementation of the Keyword Spotting Perceiver that is benchmarked with the Google Speech Commands dataset.
- **SpeechBrain's wav2vec2.0** [4] - An open-source implementation of the wav2vec2.0 model for the SpeechBrain toolkit.
- **speacher** [5] - A speech teacher framework that enables research in curriculum learning for speech recognition and quality assessment for speech synthesis models.

[2] https://zenodo.org/record/5734340

[3] https://github.com/gcambara/speech-commands

[4] https://github.com/speechbrain/speechbrain/blob/develop/recipes/LibriSpeech/self-supervised-learning/wav2vec2/train_sb_wav2vec2.py

[5] https://github.com/gcambara/speacher

- **speechbook** [6] - Recipes used to train some of our ASR models, like the Transformer model with voice quality and prosody features.
- **wav2letter** [7] - Forked version of the wav2letter++ framework that enables training with pitch and voice quality features.
- **cape** [8] - An open-source implementation of Continuous Augmented Positional Embeddings (CAPE) (Likhomanenko et al., 2021), generalizable positional embeddings for speech, audio, text and images.

6.3.4. Project Deliveries

Many research outcomes and developed models have been used for project deliveries during the time of this work. Find here these mentioned:

- **INGENIOUS** [9] - A voice-to-voice translator for European emergency teams speaking in different languages, where I deployed ASR models for Spanish, French, German and Swedish.
- **Aura** [10] - Telefónica's voice assistant, for which we delivered speech enhancement models to improve its wake-up word detection.
- **Dolby Labs** - During a six months internship I implemented a novel self-supervised learning model that I deployed for the company's internal use, as well as code for benchmarking against multiple speech downstream tasks like speech recognition, speaker diarization or speaker identification.
- **Amazon Alexa** - In the course of another six-month internship I worked on implementing a new text-to-speech paradigm based on large language models and diffusion decoders. Part of this research is the core of the new Alexa components that are currently being announced [11] and deployed to user devices. I have not included these advancements in the thesis main corpus as it went out of the scope of our work, which is mainly focused on speech recognition, and also because it is undergoing publication reviewing within the company.

[6] https://github.com/gcambara/speechbook
[7] https://github.com/gcambara/wav2letter/tree/wav2letter_pitch
[8] https://github.com/gcambara/cape
[9] https://ingenious-first-responders.eu/
[10] https://aura.telefonica.com/es/
[11] https://www.amazon.science/blog/alexa-unveils-new-speech-recognition-text-to-speech-technologies

6.3.5. Attributions

This work has been possible thanks to the funding and collaboration of many partners, which attributions are mentioned in the following list, following no particular order:

- The TALN Natural Language Processing Research Group (UPF - DTIC Department) has ensured and managed the funding of the author during the whole course of this thesis. For that, I give special thanks to Dr. Mireia Farrús and Dr. Leo Wanner.
- Particularly, most of the funding received from TALN has come from the INGENIOUS European project, which is funded by the European Union's Horizon 2020 Research and Innovation Programme and the Korean Government under Grant Agreement No 833435.
- Telefónica Research funded the first year of this thesis, first during a six-month internship and then through a partnership with TALN, for which I want to thank Dr. Jordi Luque.
- The collection and design of the "OK Aura" dataset, as well as the development of the speech enhancement model for wake-up word detection, were done in collaboration with Dr. Jordi Luque, Dr. Carlos Segura, Dr. Mireia Farrús, David Bonet, Fernando López and Pablo López.
- Dolby Labs provided funding the second year of this thesis, for a six-month internship with Dr. Joan Serrà, Dr. Santiago Pascual and Dr. Jordi Pons.
- Amazon Text-to-Speech Research granted an internship for six months in the third year of the thesis, for which I want to thank Dr. Elena Sokolova and Dr. Antonio Bonafonte.
- Thanks to Alex Peiró-Lilja, Rodolfo Zevallos, Martin Kocour and Ariadna Sánchez for being frequent collaborators on works about ASR for low-resource languages and text-to-speech assessment, speech enhancement for TV shows or curriculum learning for speech. Some of these topics have been treated in this thesis and others have been left out to narrow down the scope of it.

Bibliography

Allen, J. B. and Berkley, D. A. (1979). Image method for efficiently simulating small-room acoustics. The Journal of the Acoustical Society of America, 65(4):943–950.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al. (2016). Deep speech 2: End-to-end speech recognition in English and Mandarin. In International Conference on Machine Learning, pages 173–182. PMLR.

Ananthakrishnan, S. and Narayanan, S. S. (2007). Automatic prosodic event detection using acoustic, lexical, and syntactic evidence. IEEE transactions on audio, speech, and language processing, 16(1):216–228.

Apple, S. (2017). Hey Siri: An On-device DNN-powered Voice Trigger for Apple's Personal Assistant. `https://machinelearning.apple.com/research/hey-siri`.

Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., Morais, R., Saunders, L., Tyers, F., and Weber, G. (2020). Common voice: A massively-multilingual speech corpus. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 4218–4222.

Arik, S. O., Kliegl, M., Child, R., Hestness, J., Gibiansky, A., Fougner, C., Prenger, R., and Coates, A. (2017). Convolutional recurrent neural networks for small-footprint keyword spotting. arXiv preprint arXiv:1703.05390.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in Neural Information Processing Systems, 33:12449–12460.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Bahl, L., Bakis, R., Cohen, P., Cole, A., Jelinek, F., Lewis, B., and Mercer, R. (1981). Continuous parameter acoustic processing for recognition of a natural speech corpus. In ICASSP'81. IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 6, pages 1149–1152. IEEE.

Baum, L. E. and Eagon, J. A. (1967). An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology.

Baum, L. E. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. The annals of mathematical statistics, 37(6):1554–1563.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. The annals of mathematical statistics, 41(1):164–171.

Benba, A., Jilbab, A., and Hammouch, A. (2014). Hybridization of best acoustic cues for detecting persons with Parkinson's disease. In 2014 Second World Conference on Complex Systems (WCCS), pages 622–625. IEEE.

Berg, A., O'Connor, M., and Cruz, M. T. (2021). Keyword transformer: A self-attention model for keyword spotting. arXiv preprint arXiv:2104.00769.

Betker, J. (2023). Better speech synthesis through scaling. arXiv preprint arXiv:2305.07243.

Boersma, P. and Van Heuven, V. (2001). Speak and unspeak with Praat. Glot International, 5(9/10):341–347.

Borsos, Z., Marinier, R., Vincent, D., Kharitonov, E., Pietquin, O., Sharifi, M., Roblek, D., Teboul, O., Grangier, D., Tagliasacchi, M., et al. (2023). Audiolm: a language modeling approach to audio generation. IEEE/ACM Transactions on Audio, Speech, and Language Processing.

Bougrine, S., Cherroun, H., and Ziadi, D. (2018). Prosody-based spoken Algerian Arabic dialect identification. Procedia Computer Science, 128:9–17.

Braun, S. and Tashev, I. (2020). Data augmentation and loss normalization for deep noise suppression. arXiv preprint arXiv:2008.06412.

Bredin, H., Yin, R., Coria, J. M., Gelly, G., Korshunov, P., Lavechin, M., Fustes, D., Titeux, H., Bouaziz, W., and Gill, M.-P. (2020). pyannote.audio: neural

building blocks for speaker diarization. In ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing, Barcelona, Spain.

Cámbara, G., López, F., Bonet, D., Gómez, P., Segura, C., Farrús, M., and Luque, J. (2022). Tase: Task-aware speech enhancement for wake-up word detection in voice assistants. Applied Sciences, 12(4):1974.

Cámbara, G., Luque, J., and Farrús, M. (2020). Detection of speech events and speaker characteristics through photo-plethysmographic signal neural processing. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7564–7568. IEEE.

Campbell, N. and Mokhtari, P. (2003). Voice quality: the 4th prosodic dimension. pages 2417–2420.

Canavan, A. and Zipperlen, G. (1996). CALLHOME Spanish Speech. LDC96S35. Web Download. Philadelphia: Linguistic Data Consortium.

Carletta, J. (2007). Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. Language Resources and Evaluation, 41(2):181–190.

Chan, W., Jaitly, N., Le, Q. V., and Vinyals, O. (2015). Listen, attend and spell. arXiv preprint arXiv:1508.01211.

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., et al. (2022). Wavlm: Large-scale self-supervised pre-training for full stack speech processing. IEEE Journal of Selected Topics in Signal Processing, 16(6):1505–1518.

Chung, Y.-A. and Glass, J. (2020). Generative pre-training for speech with autoregressive predictive coding. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3497–3501. IEEE.

Collobert, R., Puhrsch, C., and Synnaeve, G. (2016). Wav2letter: an end-to-end ConvNet-based speech recognition system. arXiv preprint arXiv:1609.03193.

Cámbara, G., Luque, J., Bonet, D., López, F., Farrús, M., Gómez, P., and Segura, C. (2021). OK Aura Wake-up Word Dataset.

Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. (2017). Language modeling with gated convolutional networks. In International Conference on Machine Learning, pages 933–941. PMLR.

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(4):357–366.

de Cheveignè, A. and Kawahara, H. (2002). Yin, a fundamental frequency estimator for speech and music. The Journal of the Acoustical Society of America, 111(4):1917–1930.

Dean, D. B., Sridharan, S., Vogt, R. J., and Mason, M. W. (2010). The QUT-NOISE-TIMIT corpus for the evaluation of voice activity detection algorithms. Proceedings of Interspeech 2010.

Défossez, A., Synnaeve, G., and Adi, Y. (2020). Real time speech enhancement in the waveform domain. arXiv preprint arXiv:2006.12847.

Défossez, A., Usunier, N., Bottou, L., and Bach, F. (2019). Music source separation in the waveform domain. arXiv preprint arXiv:1911.13254.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Ephraim, Y. and Van Trees, H. L. (1995). A signal subspace approach for speech enhancement. IEEE Transactions on speech and audio processing, 3(4):251–266.

Farrús, M. and Hernando, J. (2009). Using jitter and shimmer in speaker verification. IET Signal Process. 2009; 3 (4): 247-57.

Farrús, M., Hernando, J., and Ejarque, P. (2007). Jitter and shimmer measurements for speaker recognition. In Proceedings of the Interspeech, Antwerp, Belgium.

Fiscus, J. G. (1997). A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER). In 1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings, pages 347–354.

Gales, M., Young, S., et al. (2008). The application of hidden Markov models in speech recognition. Foundations and Trends® in Signal Processing, 1(3):195–304.

Ge, F. and Yan, Y. (2017). Deep neural network based wake-up-word speech recognition with two-stage detection. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2761–2765. IEEE.

Ghahremani, P., BabaAli, B., Povey, D., Riedhammer, K., Trmal, J., and Khudanpur, S. (2014). A pitch extraction algorithm tuned for automatic speech recognition. In 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 2494–2498. IEEE.

Gong, Y., Chung, Y.-A., and Glass, J. (2021). Ast: Audio spectrogram transformer. arXiv preprint arXiv:2104.01778.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. Advances in Neural Information Processing Systems, 27.

Graff, D., Huang, S., Cartagena, I., Walker, K., and Cieri, C. (2010). Fisher Spanish Speech. LDC2010S01. DVD. Philadelphia: Linguistic Data Consortium.

Graves, A. (2012). Sequence transduction with recurrent neural networks. arXiv preprint arXiv:1211.3711.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning, pages 369–376.

Graves, A. and Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In International Conference on Machine Learning, pages 1764–1772. PMLR.

Guglani, J. and Mishra, A. (2020). Automatic speech recognition system with pitch dependent features for Punjabi language on Kaldi toolkit. Applied Acoustics, 167:107386.

Han, W., Zhang, Z., Zhang, Y., Yu, J., Chiu, C.-C., Qin, J., Gulati, A., Pang, R., and Wu, Y. (2020). ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context. Proc. Interspeech 2020, pages 3610–3614.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al. (2014). Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567.

Hannun, A., Lee, A., Xu, Q., and Collobert, R. (2019). Sequence-to-Sequence Speech Recognition with Time-Depth Separable Convolutions. In INTERSPEECH.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778.

Heafield, K. (2011). KenLM: Faster and Smaller Language Model Queries. In Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT '11, page 187–197, USA. Association for Computational Linguistics.

Hou, J., Shi, Y., Ostendorf, M., Hwang, M.-Y., and Xie, L. (2020). Mining effective negative training samples for keyword spotting. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7444–7448. IEEE.

Hsiao, R., Ma, J., Hartmann, W., Karafiát, M., Grézl, F., Burget, L., Szöke, I., Černockỳ, J. H., Watanabe, S., Chen, Z., et al. (2015). Robust speech recognition in unknown reverberant and noisy conditions. In 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pages 533–538. IEEE.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29:3451–3460.

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7132–7141.

Hu, Y., Chen, C., Li, R., Zhu, Q., and Chng, E. S. (2023). Noise-aware speech enhancement using diffusion probabilistic model. arXiv preprint arXiv:2307.08029.

Jadoul, Y., Thompson, B., and De Boer, B. (2018). Introducing parselmouth: A python interface to Praat. Journal of Phonetics, 71:1–15.

Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., and Carreira, J. (2021). Perceiver: General perception with iterative attention. In International Conference on Machine Learning, pages 4651–4664. PMLR.

Jaitly, N. and Hinton, G. (2011). Learning a better representation of speech soundwaves using restricted Boltzmann machines. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5884–5887. IEEE.

Kenton, J. D. M.-W. C. and Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT, pages 4171–4186.

Killer, M., Stüker, S., and Schultz, T. (2003). Grapheme based speech recognition. In Interspeech.

Kim, C. and Stern, R. M. (2008). Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis. In Ninth Annual Conference of the International Speech Communication Association.

Kocour, M. (2019). Automatic Speech Recognition System Continually Improving Based on Subtitled Speech Data. Diploma thesis, Brno University of Technology, Faculty of Information Technology, Brno. technical supervisor Dr. Ing. Jordi Luque Serrano. supervisor Doc. Dr. Ing. Jan Černocky.

Kocour, M., Cámbara, G., Luque, J., Bonet, D., Farrús, M., Karafiát, M., Veselỳ, K., and Černockỳ, J. (2021). BCN2BRNO: ASR System Fusion for Albayzin 2020 Speech to Text Challenge. IberSPEECH2020.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90.

Kudo, T. and Richardson, J. (2021). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. NSU.

Kumar, R., Rodehorst, M., Wang, J., Gu, J., and Kulis, B. (2020). Building a robust word-level wakeword verification network. In INTERSPEECH, pages 1972–1976.

Kumar, R., Yeruva, V., and Ganapathy, S. (2018). On Convolutional LSTM Modeling for Joint Wake-Word Detection and Text Dependent Speaker Verification. In Interspeech, pages 1121–1125.

LeCun, Y. et al. (2015). LeNet-5, convolutional neural networks. URL: http://yann. lecun. com/exdb/lenet, 20(5):14.

Li, L., Kang, Y., Shi, Y., Kürzinger, L., Watzel, T., and Rigoll, G. (2021). Adversarial joint training with self-attention mechanism for robust end-to-end speech recognition. EURASIP Journal on Audio, Speech, and Music Processing, 2021:1–16.

Li, X., Tao, J., Johnson, M. T., Soltis, J., Savage, A., Leong, K. M., and Newman, J. D. (2007). Stress and emotion classification using jitter and shimmer features. In 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, volume 4, pages IV–1081. IEEE.

Likhomanenko, T., Synnaeve, G., and Collobert, R. (2019). Who Needs Words? Lexicon-Free Speech Recognition. Proc. Interspeech 2019, pages 3915–3919.

Likhomanenko, T., Xu, Q., Synnaeve, G., Collobert, R., and Rogozhnikov, A. (2021). Cape: Encoding relative positions with continuous augmented positional embeddings. Advances in Neural Information Processing Systems, 34:16079–16092.

Liptchinsky, V., Synnaeve, G., and Collobert, R. (2017). Letter-based speech recognition with gated convnets. ArXiv, abs/1712.09444.

Liu, A. T., Yang, S.-w., Chi, P.-H., Hsu, P.-c., and Lee, H.-y. (2020). Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6419–6423. IEEE.

Liu, B., Nie, S., Liang, S., Liu, W., Yu, M., Chen, L., Peng, S., Li, C., et al. (2019). Jointly adversarial enhancement training for robust end-to-end speech recognition. In Interspeech, pages 491–495.

Lleida, E., Ortega, A., Miguel, A., Bazán, V., Pérez, C., Zotano, M., and De Prada, A. (2018). RTVE2018 Database Description.

Lleida, E., Ortega, A., Miguel, A., Bazán-Gil, V., Pérez, C., Gómez, M., and de Prada, A. (2019). Albayzin 2018 evaluation: the IberSpeech-RTVE challenge on speech technologies for Spanish broadcast media. Applied Sciences, 9(24):5412.

Lleida, E., Ortega, A., Miguel, A., Bazán-Gil, V., Pérez, C., Gómez, M., and De Prada, A. (2020). RTVE2020 Database Description.

Llombart, J., Ribas, D., Miguel, A., Vicente, L., Ortega, A., and Lleida, E. (2021). Progressive loss functions for speech enhancement with deep neural networks. EURASIP Journal on Audio, Speech, and Music Processing, 2021(1):1–16.

Loizou, P. C. (2013). Speech enhancement: theory and practice. CRC press.

Loshchilov, I. and Hutter, F. (2018). Decoupled weight decay regularization. In International Conference on Learning Representations.

Lu, Y.-J., Wang, Z.-Q., Watanabe, S., Richard, A., Yu, C., and Tsao, Y. (2022). Conditional diffusion probabilistic model for speech enhancement. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7402–7406. IEEE.

Maas, A. L., Le, Q. V., O'Neil, T. M., Vinyals, O., Nguyen, P., and Ng, A. Y. (2012). Recurrent neural networks for noise reduction in robust ASR. In Thirteenth Annual Conference of the International Speech Communication Association.

Magimai-Doss, M., Stephenson, T. A., and Bourlard, H. (2003). Using pitch frequency information in speech recognition. In 8th European Conference on Speech Communication and Technology.

Manohar, V., Povey, D., and Khudanpur, S. (2017). JHU Kaldi system for Arabic MGB-3 ASR challenge using diarization, audio-transcript alignment and transfer learning. In 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), volume 2018-, pages 346–352. IEEE.

Meyer, J. and Simmer, K. U. (1997). Multi-channel speech enhancement in a car environment using Wiener filtering and spectral subtraction. In 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 2, pages 1167–1170. IEEE.

"mhiro2", M. H. (2019). Freesound Audio Tagging 2019: Simple 2D-CNN Classifier with PyTorch. `https://www.kaggle.com/mhiro2/simple-2d-cnn-classifier-with-pytorch/`.

Michaely, A. H., Zhang, X., Simko, G., Parada, C., and Aleksic, P. (2017). Keyword spotting for Google assistant using contextual speech recognition. In 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pages 272–278. IEEE.

Mirzaei, S., El Yacoubi, M., Garcia-Salicetti, S., Boudy, J., Kahindo, C., Cristancho-Lacroix, V., Kerhervé, H., and Rigaud, A.-S. (2018). Two-stage feature selection of voice parameters for early Alzheimer's disease prediction. IRBM, 39(6):430–435.

Moore, R. K. (2003). A comparison of the data requirements of automatic speech recognition systems and human listeners. In Eighth European Conference on Speech Communication and Technology.

Moreno, A., Gedge, O., Heuvel, H., Höge, H., Horbach, S., Martin, P., Pinto, E., Rincón, A., Senia, F., and Sukkar, R. (2002). SpeechDat across all America: SALA II.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. arXiv preprint arXiv:1904.01038.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). LibriSpeech: an ASR corpus based on public domain audio books. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5206–5210. IEEE.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019). SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. Proc. Interspeech 2019, pages 2613–2617.

Park, S. R. and Lee, J. (2016). A fully convolutional neural network for speech enhancement. arXiv preprint arXiv:1609.07132.

Pascual, S., Bonafonte, A., and Serra, J. (2017). SEGAN: Speech enhancement generative adversarial network. arXiv preprint arXiv:1703.09452.

Pascual, S., Ravanelli, M., Serrà, J., Bonafonte, A., and Bengio, Y. (2019). Learning problem-agnostic speech representations from multiple self-supervised tasks. Proc. Interspeech 2019, pages 161–165.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. arxiv preprint. arXiv preprint arXiv:1802.05365.

Phan, H., McLoughlin, I. V., Pham, L., Chén, O. Y., Koch, P., De Vos, M., and Mertins, A. (2020). Improving GANs for speech enhancement. arXiv preprint arXiv:2001.05532.

Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohammadi, M., and Khudanpur, S. (2018). Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks. In Proceedings of Interspeech, pages 3743–3747.

Povey, D. et al. (2011). The Kaldi speech recognition toolkit. IEEE Workshop on Automatic Speech Recognition and Understanding.

Prabhavalkar, R., Rao, K., Sainath, T. N., Li, B., Johnson, L., and Jaitly, N. (2017). A comparison of sequence-to-sequence models for speech recognition. In INTERSPEECH, pages 939–943.

Pratap, V., Hannun, A., Xu, Q., Cai, J., Kahn, J., Synnaeve, G., Liptchinsky, V., and Collobert, R. (2018). wav2letter++: The fastest open-source speech recognition system. arXiv preprint arXiv:1812.07625.

Pratap, V., Xu, Q., Kahn, J., Avidov, G., Likhomanenko, T., Hannun, A., Liptchinsky, V., Synnaeve, G., and Collobert, R. (2020). Scaling Up Online Speech Recognition Using ConvNets. Proc. Interspeech 2020, pages 3376–3380.

Rabiner, L. and Juang, B. (1986). An introduction to hidden Markov models. IEEE ASSP Magazine, 3(1):4–16.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. In International Conference on Machine Learning, pages 28492–28518. PMLR.

Raju, A., Panchapagesan, S., Liu, X., Mandal, A., and Strom, N. (2018). Data augmentation for robust keyword spotting under playback interference. arXiv preprint arXiv:1808.00563.

Rao, K., Sak, H., and Prabhavalkar, R. (2017). Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer. In 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pages 193–199. IEEE.

Ravanelli, M. and Bengio, Y. (2018). Speaker recognition from raw waveform with SincNet. In 2018 IEEE spoken language technology workshop (SLT), pages 1021–1028. IEEE.

Ravanelli, M., Brakel, P., Omologo, M., and Bengio, Y. (2016). Batch-normalized joint training for DNN-based distant speech recognition. In 2016 IEEE Spoken Language Technology Workshop (SLT), pages 28–34. IEEE.

Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., et al. (2021). SpeechBrain: A general-purpose speech toolkit. arXiv preprint arXiv:2106.04624.

Ravanelli, M., Zhong, J., Pascual, S., Swietojanski, P., Monteiro, J., Trmal, J., and Bengio, Y. (2020). Multi-task self-supervised learning for robust speech recognition. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6989–6993. IEEE.

Rethage, D., Pons, J., and Serra, X. (2018). A Wavenet for speech denoising. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5069–5073. IEEE.

Rix, A. W., Beerends, J. G., Hollier, M. P., and Hekstra, A. P. (2001). Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs. In 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221), volume 2, pages 749–752. IEEE.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 234–241. Springer.

Rouas, J.-L. (2007). Automatic prosodic variations modeling for language and dialect discrimination. IEEE Transactions on Audio, Speech, and Language Processing, 15(6):1904–1911.

Sainath, T. N. and Parada, C. (2015). Convolutional neural networks for small-footprint keyword spotting. In Sixteenth Annual Conference of the International Speech Communication Association.

Schindler, A., Mozzanica, F., Vedrody, M., Maruzzi, P., and Ottaviani, F. (2009). Correlation between the voice handicap index and voice measurements in four groups of patients with dysphonia. Otolaryngology–Head and Neck Surgery, 141(6):762–769.

Schneider, S., Baevski, A., Collobert, R., and Auli, M. (2019). wav2vec: Unsupervised pre-training for speech recognition. arXiv preprint arXiv:1904.05862.

Scholefield, M. D. (2019). Mycroft Precise. https://github.com/MycroftAI/mycroft-precise.

Schuster, M. and Nakajima, K. (2012). Japanese and Korean voice search. In 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5149–5152. IEEE.

Seo, D., Oh, H.-S., and Jung, Y. (2021). Wav2KWS: Transfer learning from speech representations for keyword spotting. IEEE Access, 9:80682–80691.

Shan, C., Zhang, J., Wang, Y., and Xie, L. (2018). Attention-based end-to-end models for small-footprint keyword spotting. arXiv preprint arXiv:1803.10916.

Slyh, R. E., Nelson, W. T., and Hansen, E. G. (1999). Analysis of mrate, shimmer, jitter, and F/sub 0/contour features across stress and speaking style in the SUSAS database. In 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258), volume 4, pages 2091–2094. IEEE.

Song, X., Wang, G., Huang, Y., Wu, Z., Su, D., and Meng, H. (2020). Speech-XLNet: Unsupervised Acoustic Model Pretraining for Self-Attention Networks. Proc. Interspeech 2020, pages 3765–3769.

Strauss, M. and Edler, B. (2021). A flow-based neural network for time domain speech enhancement. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5754–5758. IEEE.

Sun, C., Myers, A., Vondrick, C., Murphy, K., and Schmid, C. (2019). VideoBERT: A joint model for video and language representation learning. In Proceedings of the IEEE/CVF international conference on computer vision, pages 7464–7473.

Synnaeve, G., Xu, Q., Kahn, J., Likhomanenko, T., Grave, E., Pratap, V., Sriram, A., Liptchinsky, V., and Collobert, R. (2019). End-to-end ASR: from supervised to semi-supervised learning with modern architectures. arXiv preprint arXiv:1911.08460.

Taal, C. H., Hendriks, R. C., Heusdens, R., and Jensen, J. (2010). A short-time objective intelligibility measure for time-frequency weighted noisy speech. In 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 4214–4217. IEEE.

Talkin, D. and Kleijn, W. B. (1995). A robust algorithm for pitch tracking (RAPT). Speech Coding and Synthesis, 495:518.

Tang, R. and Lin, J. (2017). Honk: A PyTorch reimplementation of convolutional neural networks for keyword spotting. arXiv preprint arXiv:1710.06554.

Tang, R. and Lin, J. (2018). Deep residual learning for small-footprint keyword spotting. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5484–5488. IEEE.

Valentini Botinhao, C., Wang, X., Takaki, S., and Yamagishi, J. (2016). Speech enhancement for a noise-robust text-to-speech synthesis system using deep recurrent neural networks. In Proceedings of Interspeech 2016, Interspeech, pages 352–356. International Speech Communication Association. Interspeech 2016 ; Conference date: 08-09-2016 Through 12-09-2016.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.

Wang, C., Chen, S., Wu, Y., Zhang, Z., Zhou, L., Liu, S., Chen, Z., Liu, Y., Wang, H., Li, J., et al. (2023). Neural codec language models are zero-shot text to speech synthesizers. arXiv preprint arXiv:2301.02111.

Wang, C., Tang, Y., Ma, X., Wu, A., Okhonko, D., and Pino, J. (2020a). fairseq s2t: Fast speech-to-text modeling with fairseq. arXiv preprint arXiv:2010.05171.

Wang, G., Rosenberg, A., Chen, Z., Zhang, Y., Ramabhadran, B., Wu, Y., and Moreno, P. (2020b). Improving speech recognition using consistent predictions on synthesized speech. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7029–7033. IEEE.

Wang, Y., Deng, X., Pu, S., and Huang, Z. (2017). Residual convolutional CTC networks for automatic speech recognition. arXiv preprint arXiv:1702.07793.

Wang, Y., Mohamed, A., Le, D., Liu, C., Xiao, A., Mahadeokar, J., Huang, H., Tjandra, A., Zhang, X., Zhang, F., et al. (2020c). Transformer-based acoustic modeling for hybrid speech recognition. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6874–6878. IEEE.

Warden, P. (2018). Speech commands: A dataset for limited-vocabulary speech recognition. arXiv preprint arXiv:1804.03209.

Weiss, R. J., Chorowski, J., Jaitly, N., Wu, Y., and Chen, Z. (2017). Sequence-to-sequence models can directly translate foreign speech. arXiv preprint arXiv:1703.08581.

Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Le Roux, J., Hershey, J. R., and Schuller, B. (2015). Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. In International Conference on Latent Variable Analysis and Signal Separation, pages 91–99. Springer.

Wittig, F. and Müller, C. (2003). Implicit feedback for user-adaptive systems by analyzing the users' speech. In Proceedings of the Workshop on Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen (ABIS), Karlsruhe, Germany.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv preprint arXiv:1910.03771.

Xu, Y., Du, J., Dai, L.-R., and Lee, C.-H. (2013). An experimental study on speech enhancement based on deep neural networks. IEEE Signal Processing Letters, 21(1):65–68.

Yadav, I. C., Shahnawazuddin, S., and Pradhan, G. (2019). Addressing noise and pitch sensitivity of speech recognition system through variational mode decomposition based spectral smoothing. Digit. Signal Process., 86:55–64.

Yamamoto, R., Song, E., and Kim, J.-M. (2020). Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6199–6203. IEEE.

Yamamoto, T., Nishimura, R., Misaki, M., and Kitaoka, N. (2019). Small-Footprint Magic Word Detection Method Using Convolutional LSTM Neural Network. In INTERSPEECH, pages 2035–2039.

Yang, L.-P. and Fu, Q.-J. (2005). Spectral subtraction-based speech enhancement for cochlear implant patients in background noise. The journal of the Acoustical Society of America, 117(3):1001–1004.

Yang, S.-w., Chi, P.-H., Chuang, Y.-S., Lai, C.-I. J., Lakhotia, K., Lin, Y. Y., Liu, A. T., Shi, J., Chang, X., Lin, G.-T., et al. (2021). SUPERB: Speech processing universal performance benchmark. arXiv preprint arXiv:2105.01051.

Yang, X., Audhkhasi, K., Rosenberg, A., Thomas, S., Ramabhadran, B., and Hasegawa-Johnson, M. (2018). Joint modeling of accents and acoustics for multi-accent speech recognition. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. Advances in Neural Information Processing Systems, 32.

Yoon, K.-M. and Kim, W. (2020). Small-footprint wake up word recognition in noisy environments employing competing-words-based feature. Electronics, 9(12):2202.

Youden, W. J. (1950). Index for rating diagnostic tests. Cancer, 3(1):32–35.

Zeghidour, N., Usunier, N., Kokkinos, I., Schaiz, T., Synnaeve, G., and Dupoux, E. (2018a). Learning filterbanks from raw speech for phone recognition. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5509–5513. IEEE.

Zeghidour, N., Usunier, N., Synnaeve, G., Collobert, R., and Dupoux, E. (2018b). End-to-end speech recognition from the raw waveform. Proc. Interspeech 2018, pages 781–785.

Zeghidour, N., Xu, Q., Liptchinsky, V., Usunier, N., Synnaeve, G., and Collobert, R. (2018c). Fully convolutional speech recognition. arXiv preprint arXiv:1812.06864.

Zewoudie, A. W., Luque, J., and Hernando, J. (2014). Jitter and shimmer measurements for speaker diarization. In VII Jornadas en Tecnología del Habla and III Iberian SLTech Workshop, pages 21–30, Las Palmas de Gran Canaria, Spain.

Zheng, Y., Sproat, R., Gu, L., Shafran, I., Zhou, H., Su, Y., Jurafsky, D., Starr, R., and Yoon, S.-Y. (2005). Accent detection and speech recognition for Shanghai-accented Mandarin. In Ninth European Conference on Speech Communication and Technology.

Zorilă, C., Boeddeker, C., Doddipatla, R., and Haeb-Umbach, R. (2019). An investigation into the effectiveness of enhancement in ASR training and test

for CHiME-5 dinner party transcription. In 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pages 47–53. IEEE.